August 1992

ISSN 0956-9979

VIRUS BULLETIN

THE AUTHORITATIVE INTERNATIONAL PUBLICATION
ON COMPUTER VIRUS PREVENTION,
RECOGNITION AND REMOVAL

Editor: **Edward Wilding**

Technical Editor: **Fridrik Skulason**

CONTENTS

EDITORIAL

The Copyright Gauntlet 2

VIRUS PREVALENCE TABLES 3

INSIGHT

Extracts From The Hell Pit 3

LEGAL ISSUES

Virus Searching and Copyright 5

IBM PC VIRUSES (UPDATE) 7

DIRTY MACS

T4-A and T4-B 11

KNOWN APPLE MAC VIRUSES 12

TUTORIAL

Multi-partite and Companion Viruses 16

PC VIRUS ANALYSES

1. Helloween 17

2. Palestinian (aka MSJ) 19

PRODUCT REVIEWS

1. *Integrity Master* 21

2. *Vi-Spy* - Professional Edition 24

BOOK REVIEW

Approaching Zero 27

END-NOTES & NEWS 28

EDITORIAL

The Copyright Gauntlet

The search strings published regularly in *VB* are not copyrighted by *Virus Bulletin Ltd* and their widespread use in virus scanning programs is encouraged subject to the normal caveats regarding false positives, typographic errors and the other potential hazards of non-detection or misidentification of virus code.

The traditional arguments over copyright infringement have arisen internally within the anti-virus industry and bellicose insinuations, threats and warnings regarding the sanctity of virus detection patterns continue, even to this day, to rumble from various quarters.

The exact ownership of virus detection strings is unclear and the issue of their copyright amounts to a legal minefield for any company seeking to litigate against a competitor should it copy detection patterns without authorisation, either in part or *in toto*.

In light of this veiled industry animosity, it is amusing to see that virus writer 'Nowhere Man' (whose latest activities are reported on pages 3-4) has entered the fray and thrown down his own gauntlet over this issue. In a display of quite breathtaking temerity, 'Nowhere Man' has made it clear that he regards any virus assembled with his *Virus Creation Laboratory[TM]* (note the impertinent trademark) as copyright and threatens legal retribution should a manufacturer seek to detect any resulting virus specimen using an extracted hexadecimal search pattern.

The complexities surrounding computer virus detection and copyright are discussed in this issue by barrister-at-law Owen Keane, himself a former specialist in copyright matters. In the absence of a precedent case, Mr Keane concludes that a court of law would show little sympathy to a virus writer claiming copyright. Instinctively it is difficult to conclude otherwise; one can hardly imagine the anti-virus industry quaking in its boots at the prospect of 'Nowhere Man' issuing writs left, right and centre.

We await a test-case with interest. Whether Nowhere Man's writ lands first on the desk of *IBM Corporation*, *Symantec Inc*., *Fifth Generation System*s or *Central Point Software*, we must wait and see. The inescapable conclusion is that it'll get him nowhere, man.

Editorial Policy - Virus Analyses

The in-depth computer virus analyses published in *VB* over the last seven months have concentrated almost exclusively on those specimens found in the wild, i.e. those viruses which are causing 'real world' disruption. This policy is based on the premise that *VB's* readership requires detailed information only about those threats which are imminent or likely to be encountered and that expending time-consuming analysis on remote or unlikely threats, ineffectual code or laboratory exercises is wasteful and unnecessary.

The virus prevalence tables, which show those viruses currently in circulation in the United Kingdom, have served as a guide to which specimens merit this detailed analysis. Exceptions to this rule include innovative viruses which have a significant impact on detection methods and/or recovery. The Mutation Engine is one such example.

To make technical information more accessible, summary tables appear at the end of each written report.

OS/2 Virus Seen In Night Sky

The recent publication of a brutally short and unsubstantiated item in the UK magazine *PC Week* has caused a wave of speculation and rumour. The report (known in the trade as a 'nib' or 'news in brief') read as follows:

> ***Virus Support***
>
> *Who says no one is writing for OS/2? According to Ray O'Connell of S&S International, two viruses for the operating system are already in circulation.*

It is the editorial 'silly season', admittedly, but surely crop circles, flying saucers and sightings of Elvis (all of which are far more prevalent than the stated phenomena) would have sufficed as back-page filler! Anxious telephone enquiries were received by *VB* within hours of *PC Week* 'hitting the streets'. Predictably, the mystery of these OS/2 viruses gradually unravelled itself as a sorry tale of lazy journalism and 'crossed lines'.

According to Iolo Davidson of *S&S International* what Mr O'Connell *really* meant to say was that two people had *claimed* to write the first OS/2 virus; claims apparently disbelieved by *S&S* supremo Alan Solomon who has, however, mentioned the fact that the claims have been made to a number of people.

With the enquiries that *PC Week's* inaccurate, alarmist report generated, it is indeed fortuitous that *S&S International* has an *OS/2 Anti-Virus Toolkit available* from stock (tel +44 0442 877877, all major credit cards accepted).

In the meantime, OS/2 users can rest easy in their beds. At the moment there are no substantiated reports of OS/2 viruses in the laboratory, in the wild or from other galaxies. But as Mr Davidson is quick to point out, 'Writing an OS/2 virus is a trivial process. Someone will do it eventually.'

Virus Prevalence Table - May 1992

Incidents reported to VB in the UK during May 1992

Virus	Incidents	(%) Reports
Form	11	28.2%
New Zealand II	6	15.4%
Cascade	5	12.8%
Tequila	3	7.7%
Spanish Telecom	3	7.7%
Michelangelo	2	5.1%
Nomenklatura	2	5.1%
Jerusalem	2	5.1%
Keypress	2	5.1%
Vienna II	1	2.5%
NoInt	1	2.5%
Yankee 44	1	2.5%
Total	39	100%

Virus Prevalence Table - June 1992

Incidents reported to VB in the UK during June 1992

Virus I	Incidents	(%)Reports
New Zealand II	11	21%
Form	10	19%
Tequila	7	13.4%
Vacsina	4	7.7%
Michelangelo	3	5.7%
Jerusalem	2	3.8%
1575	2	3.8%
Flip	2	3.8%
Cascade	2	3.8%
Helloween	1	1.9%
Dark Avenger	1	1.9%
Nomenklatura	1	1.9%
SBC	1	1.9%
W-13	1	1.9%
Spanish Telecom II	1	1.9%
4K	1	1.9%
Joshi	1	1.9%
Disk Killer	1	1.9%
Total	52	100%

INSIGHT

Extracts From The Hell Pit

Since the temporary closure of Todor Todorov's notorious virus exchange (VX) bulletin board system in Sofia (due to electricity shortages rather than official action), *The Hell Pit* BBS based in California has become the most active and dangerous VX in the world. Hundreds of live virus specimens and source code files are arrayed ready for download.

Logging onto this board is simplicity itself; an assumed ID with a plausible contact reference gains any inquisitive browser unrestricted access along with unlimited upload and download rights. The board is run by the self-styled *Phalcon/Skism* group and one of its most prolific contributors is 'Nowhere Man' of the *NuKe* virus writing circle.

The Virus Creation Laboratory

On July 5th 1992 Nowhere Man announced the availability of his 'Virus Creation Laboratory' (VCL). This is a simple menu-driven virus construction set with on-line help with which to assemble a variety of virus programs. The 'documentation' describes the objectives of program: 'No longer does one need to spend weeks writing and debugging assembly language to produce a working, competitive virus. With V.C.L. all of the work is done for you - you just choose the options and effects of the virus, and it does the rest, leaving you free to experiment with different effects and concentrate on creativity. What was once a matter of hours, days, or even weeks is reduced to a few minutes in the slick V.C.L.' The author boasts that his program will 'redefine the virus-writing community'.

Observed Code Effects

Somewhat disappointingly, after all these claims, the viruses which this construction set generates are extremely primitive non-resident COM file infectors (overwriting or appending) which often fail to restore control to their host program. The viruses are encrypted so no two samples which are generated are alike, but (crucially) they are not self-modifying, i.e. reliable search patterns can be extracted from any particular VCL generated virus.

Intriguing Documentation

The danger which this program poses is minimal; indeed the VCL (in its present form) can be discounted as a serious threat. The really *interesting* aspect of the program resides in its documentation which is altogether far more intriguing than any of the code which the program generates.

Extracts from the documentation which accompanies the VCL provide an interesting insight into the mind of the virus writer and in particular this author's seemingly defensive attitude regarding issues of copyright and reverse engineering. It is probable that the following 'legalese' is simply a parody of commercial software warranties, although the author inadvertently raises some important copyright issues!

> Virus Creation Laboratory
> Version 1.00
>
> Copyright (c) 1992 Nowhere Man and [NuKE] WaReZ V.C.L. and all documentation written by Nowhere Man [NuKE] and [NuKE] WaReZ are trademarks of [NuKE] International Software Development Corperation. Borland C++, Turbo Assembler, and Turbo Linker are registered trademarks of Borland International.
>
> Microsoft is a registered trademark of Microsoft Corporation. Microsoft: Proud to bring you ten years of the 640k limit.
>
> Legalese
> --------
>
> Nowhere Man and [NuKE] WaReZ are hereby not responsible for any damages caused by the use or misuse of Nowhere Man's Virus Creation Laboratory (V.C.L.) nor by the use or misuse of any program produced, in whole or in part, by V.C.L. The author, Nowhere Man, will not be held responsible for any losses incurred, either directly or indirectly, by the use of this product or by the use of any program generated, in whole or in part, by this product. This product is distributed 'as is' with no warranties expressed or implied. Use this product entirely at your own risk. The author makes no guarantees as to the correct functioning of this product. The author reserves the right to make modifications at any time without prior notice.

The explicit declaration that the VCL program is used at the user's own risk is an attempt to deny culpability. A not dissimilar warranty which accompanied the AIDS Information Diskette (*VB*, January 1990, p.10) was seen by some lawyers as a possible defence against prosecution.

> All code produced, in whole or in part, by Nowhere Man's Virus Creation Laboratory (V.C.L.) automatically becomes the sole property of Nowhere Man and [NuKE] WaReZ. All binary code produced from assembler source code generated in whole or in part by V.C.L. likewise becomes the sole property of Nowhere Man and [NuKE] WaReZ. Any use of such code, in whole or in part, for the purpose of inclusion in a product, commerical or otherwise, designed to detect or eliminate said code on an electronic medium is expressly forbidden without the full written consent of Nowhere Man and [NuKE] WaReZ. This includes, but is not limited to, virus detection and removal programs, CHK4BMB-type products or other products designed to detect potentially damaging code within programs, and programs designed to detect the presence of a sequence of binary data within a computer program.
> Source and binary code produced by V.C.L. may be freely distributed and studied, so long as such distribution and research is not for the purpose of examining said code to determine weaknesses and/or methods of detection and/or removal on an electronic medium.
>
> Any reverse-engineering, disassembly, or other attempts to determine the nature of code known to be produced by V.C.L. for purposes such as those enumerated above is likewise expressly forbidden without the full written consent of Nowhere Man and [NuKE] WaReZ.

Inevitably, the VCL viruses *will* be disassembled and corresponding search data extracted for inclusion in search engines (be it in the form of specific hexadecimal patterns or a generic detection algorithm to detect all progeny of the VCL). *Virus Bulletin's* Technical Editor is currently analysing the VCL and has thus already contravened the terms of the 'warranty', a pattern which will be formally set with the publication of more detailed information about the VCL viruses in next month's edition of *VB*.

The virus writer is equally keen to retain accreditation for his efforts and requests that users of the VCL do not remove his 'OEM label'!

> When distributing virii, trojans, or logic bombs created with V.C.L., please give credit to Nowhere Man's Virus Creation Laboratory. Editing out the [VCL] marker in virii is a no-no. It's five lousy bytes. I spent months on this project, the least you can do is give me some credit.

Nowhere Man offers 'technical support' via *The Hell Pit* and actively encourages users to report bugs and provide constructive criticism. His promised forthcoming attractions include an appending .EXE infector, 'Virex-Protection(C)' ('defeats all TSR anti-virus products'), and 'Cryptex(C)' (a polymorphic encryption scheme).

Finally, acknowledgements are given to a number of minor virus writing luminaries and to the following individuals most of whom are more readily identifiable!

> Jeers go out to John [McAfee], Ross [Greenberg], Pat [Hoffman], Aryeh [Goretsky - McAfee Associates], Vesselin [Bontchev], Dennis [Steinauer - NIST?], Paul [?], and any others who profit off our work. This should more than keep you busy for a while... A special 'Fuck You' to James Dahan, a.k.a Fat Cat (must be pretty fat since he's a one-man 'vigilante' group!). Go back to the litter box that you crawled out of.

LEGAL ISSUES

Owen Keane

Virus Searching and Copyright

When considering the proprietary, or otherwise, nature of 'search strings' published by *VB* and used by scanning programs, a number of issues arise, some of which are too complex to discuss fully. However, a major consideration will be what copyrights can or do exist. The two main issues are: whether the scanners are in breach of copyright; and whether scanners have a copyright of their own which can be breached by competitors copying search strings.

'That which is worth copying is worth protecting' [1] is a key if well worn phrase. UK copyright law only protects the *expression* of ideas rather than ideas *themselves*. Computer programs are included in the definition of literary works [2] for copyright law purposes, and are entitled to the same protection as books and journals. That protection prohibits copying or adapting a work completely, or doing either in relation to a substantial part of it, whether directly or indirectly [3]. The critical expression here is 'substantial', which is not defined statutorily, instead being left to interpretation by the courts.

Difficulties persist in the courts' dealings with computer matters, the reasons being expressed as 'those who live by words such as judges and lawyers, find it difficult to communicate adequately or receive communication adequately from those who live by a different system of discipline based upon mathematics and electronics' [4]. Computer matters have little direct equivalent in the literary world. This fact comes to the fore when deciding whether copyright exists or is infringed by copying search strings.

Qualifying for Copyright

Copyright arises automatically in original literary works once published, subject to a few restrictions. These include the work being a minimum size, and a minimum level of effort being used in its creation; in the past single words and short phrases/titles [5], and even a simple drawing (albeit on policy grounds), have been denied copyright protection.

Prima facie a virus is like any other program, a literary work, and should attract copyright providing the author is a qualifying person [6], and it was itself not a breach of another's copyright. Despite often being short in terms of code length it is undoubtable that skill, labour and judgement are used in their creation. Arguably the virus should be excluded from copyright on grounds either of policy (public interest) or that it has been released into the public domain in a way disclaiming copyright. The act only protects author's moral rights and does not make provision for circumstances where protection should be denied.

Substance and Infringement

The search strings are portions of virus code, chosen to identify the individual virus. The chosen string is commonly a low number of bytes being a fraction of the virus.

Nevertheless the characteristic of the string makes it an important portion, arguably giving it a disproportionate significance compared to other parts. The significance is that while copyright can be breached by wholesale copying, the use of the expression 'substantial' means that a qualitative test is also used. Thus as the signature characteristic makes the string valuable, it may qualify it as a substantial part. Arguably, this could theoretically make it a breach of copyright. A further consideration could be what the chosen code actually does in terms of instructions; if significant this could affect the question of substantiality.

Scanners and Strings

A response to this could be in the argument that the code chosen does little or no more than identify the virus and so is not used as anything more than a basis to work on, like a parody, where 'the parodist must be permitted sufficient latitude to cause his reader or viewer to 'recall or conjure up' the original work if the parody is to be successful.' [7] Similarly the mere fact that a new work has been derived (not copied) although having its origins in a copyright work will not be an infringement. Further it has been said that the test of substantiality may vary depending on the type of work in question [8], e.g. where an author intends to convey information to add to the sum total of human understanding, a wider intention may be presumed for its use than for other works, to prevent it becoming sterile. [9]

The search string chosen is the product of significant skill, judgement and labour in itself but is still likely (here presumed) to be virus code originally and so not an original work, whether it is converted from a language or not. If the string has any information added to it which may not appear in the code, e.g. identifying where to look for it; memory location or disk position, or other new matters it may be said that a further amount of work has been done to it which makes it sufficiently different to be an original work of its own. As an excerpt, sufficient work must have been done on the original to impart to the product (excerpt) a quality or character it did not possess, differentiating it from the raw material. [10] Certainly the overall scanner program is a new work but it may still be tainted by the copyright code.

Here the programs do little more than use the code to identify the virus and thereafter on the basis of the identity advise on a course of action. The strings may be the basis of a new program (work) but arguably no one string is that basis alone, it is the *collection*. If the original matter were not copyright then it is possible that a new copyright may exist in the individual strings, again subject to the need for originality, substance and form obtained by additional work. This is an area of conflicting interest and judgement.

Rival Products

There may be no difficulty in two companies' scanner programs using the same string if they arrived at the string independently. Each is 'fully entitled to make use of any information ... which [is] ... available to them in the public domain... but they [are] not allowed to copy... thereby making use of the [the other's] skill and judgement and saving themselves the trouble, and very possibly the cost, of assembling their own information.'[11] Clearly 'short-cuts' are not allowed. Similarly the economic advantage of having viral code can be monopolised, as 'there is nothing in the Act[12] which gives the public at large the right to copy a compilation merely because the information contained [therein] is not available from any other source.'[13]

Here another aspect of the virus 'industry' may affect matters; the difficulty in obtaining viral code and the importance of being seen to have an up to date product are perhaps more important than in many other situations. The value of having a search pattern for a new virus may affect sales substantially, and so the string is arguably more valuable when the virus is first released than some years later. Given that the quantity 'copied' remains constant the test of quality may arguably be measured in terms of economic value rather than any other way. [14] Normally the length necessary to qualify as 'substantial' is wholly dependent on each set of facts, but need not be very large, equally it need not be *exact* copying, adaptation may suffice if substantial objective similarities exist which are not otherwise explainable. [15]

Defences

The best argument against copyright in the virii is their potential to do harm and the unconscionability in their having copyright. *VB*, for example, could avail itself of a defence under the *Copyright Designs & Patent Act* 1988. *VB* could assert that the information was treated in accord with the requirements of fair dealing for criticism or review purposes. [16] Another area may be the absence of any intention to take for the purpose of saving labour (*animus furandi*). [17]

In any case it could be hoped that the potential interests of the virus author would be overridden by equity, which has been a powerful ally in the past for those wrongly infringed. In other jurisdictions the law has striven to protect software companies[18] sometimes to the point of criticism. [19]

Compilations and Copyright

If the extraction process is repeated for each virus and the strings then are compiled into a table for use by the scanning program a new independent copyright in the compilation itself may arise, if the compilation is the result of sufficient skill, labour and judgement. As new strings are added a new copyright in the compilation will arise as it is substantially altered. [20] This copyright may be infringed by the copying of sufficient of the compilation to merit substantial taking. However, differing opinions have been expressed over what parts of a compilation attract copyright. In the past elements which were not copyright in their own have been denied copyright as part of the compilation. French courts have indicated that the compilation of an index of articles with a brief quotations to indicate their nature is permissible while in America the page numbering itself of court transcripts was copyright. [22]

Conclusion

The position is uncertain. Any case will depend on its individual facts which makes predicting the outcome difficult and deriving a rule therefrom more so. However, one would expect any virus author who tried to assert copyright to receive little sympathy or help from the courts, one would hope equity or judicial concern to prevail over even a cast iron case. If this were so then virii and their constituent strings may be in the 'public domain'.

[1] Peterson J. in *University of London Press* v *University Tutorial Press* (1916) 2 Ch. 60.

[2] S. 3(1)(b) of the *Copyright Designs & Patents Act* 1988 (*CDPA*'88).

[3] S. 16(1)(a) & (e), and 16(3), 17, 21 *CDPA*'88.

[4] *Harman J. Dun & Bradstreet Ltd* v *Typesetting Facilities Ltd*, 1992 F.S.R. 320, @ 324.

[5] *Francis Day & Hunter* v *Twentieth Century Fox Corp'n Ltd* [1940] AC 112, @ 123.

[6] Sections 153 et seq. *CDPA*'88.

[7] *Williamson Music* v *Pearson Partnership* [1987] F.S.R. 97, Judge Paul Baker QC

[8] *Copinger & Skone James on Copyright*, 1991, 13th Ed'n. @ 8-29, p.176.

[9] ibid

[10] *Macmillan & Co.* v *Cooper* (1923) 40 TLR 186, Atkinson L.J. (abridged.).

[11] *Elanco Products Ltd* v *Mandops (Agrochemical Specialists) Ltd* [1979] F.S.R. 46, @ 57 adapted

[12] The Act referred to is the 1956 Act, but the remark remains true today.

[13] *ITP Ltd & BBC Ltd* v *Time Out Ltd*. 1984 F.S.R. 64, Whitford J.

[14] ibid, infra @ pp.73 & 74.

[15] *MS Associates* v *Power* [1988] F.S.R. 242, here about 43 lines from 9,000 were exact, but there were also other factors.

[16] S.30 *CDPA* '88.

[17] See note 8, @ 8-28, p.175

[18] *Autodesk Inc. & Another* v *Martin Patrick Dayson & Others*, HC Aus, 1992.

[19] See criticism of decision ibid by Peter Presscott [1992] EIPR 189.

[20] See Harman J. in *Dun & Bradstreet Ltd* v *Typesetting Facilities* 1992 [F.S.R.] @ 325 for edition copyright difficulties.

[21] *Societé Microfar* v *Sarl 'Le Monde'* [1988] FSR 519, and *West Publishing Co.* v *Mead Data Central Inc.* [1986] 799 F2d 1219, both under different laws.

IBM PC VIRUSES (UPDATE)

Updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 21 July 1992. Entries consist of the virus' name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus using the 'search' routine of a disk utility, or preferably a dedicated scanner which contains an updatable pattern library.

Type Codes

C = Infects COM files **E** = Infects EXE files **D** = Infects DOS Boot Sector (logical sector 0 on disk)

M = Infects Master Boot Sector (Track 0, Head 0, Sector 1) **N** = Not memory-resident

R = Memory-resident after infection **P** = Companion virus **L** = Link virus

Seen Viruses

302, **439** (temporary names) - CR: Two small viruses, 302 and 439 bytes long, which do not seem to do anything other than replicate.

```
302             9C86 E03D 4B00 86E0 740A 80FC FF75 6BB8 6606 9DCF 5053 5152
439             80FC 3D74 0580 FC4B 751F 8BFA 1E07 B980 00FC F2AE E313 2681
```

AIDS-fiis - CN: A 13952 byte version of this primitive, overwriting virus, which contains the text 'eN INFeCTEe BY fiis! SAC!!'. Detected with the previously published AIDS virus pattern.

Andryushka - CER: Two polymorphic, variable-length viruses from Russia. No search pattern is possible.

Ash - CN: This 280 byte virus does nothing but replicate.

```
Ash             8DB6 0401 BF00 01B9 0400 FCF3 A4B4 1A8D 961C 02CD 21B4 4E8D
```

Astra-976 - CR: A Russian, encrypted virus with unknown effects, but it contains the text '(C) AsTrA, 1991'. One 1010 byte variant is also known, which is also able to infect EXE files.

```
Astra-976       1E06 538C C88E D88E C0BE 0B00 03F5 8BFE B984 018B DDFC AD2E
Astra-1010      1E06 5356 57FA 8CC8 8ED8 8EC0 BE78 0003 F58B FEB9 8B01 8BDD
```

Atas-384, **Atas-400** - CN: Two Polish viruses. Awaiting analysis.

```
Atas-384        A4B9 0001 58FF E1B9 2B00 B2AA 8DBE 6200 FEC2 3015 47E2 F9C3
Atas-400        948A 00B9 0800 CD21 7267 81BC 8A00 4D5A 7408 81BC 8D00 4154
```

Backfont-900 - ER: Very similar to the 905 byte variant originally reported as '905' (now Backfont-905) and detected with the same pattern.

Baobab - ER: A 1635 byte virus. Awaiting analysis.

```
Baobab          62CD 214B 8EDB BB03 008B 072D A000 8907 2EA1 5F06 8ED8 2EA1
```

BFD - MDER: There are two variants of this multi-partite virus, which is of East-European (probably Russian) origin. The virus inserts itself in unused space in the EXE file, between the file header and the actual program code. The virus stores the original boot sector in the last sector of the root directory (on 360 K diskettes) or on Head 0, Track 0, Sector 12 (hard disks).

```
BFD             BE00 7CFA 8ED0 8BE6 FB50 5656 8ED8 A113 0448 4883 2E13 0404
```

Black Jec-Sad - CN: 301 bytes, detected with the Bljec pattern. Displays a text message in September.

Blaze, **MSK** - Two closely related, badly written overwriting viruses, which contain destructive code.

```
Blaze           CD21 B43E CD21 B44F BA00 01CD 21BA 9E00 BF9E 00B0 00B9 0C00
MSK             CD21 B43E CD21 B001 B901 00B4 43CD 21B4 4FBA 0001 CD21 BA9E
```

Cascade-1701-D - CR: Minor variant of the Cascade-1701-A virus, with the encryption routine changed slightly.

```
Cascade-1701-D  012E F687 2A01 0174 0F8D B74D 01BF 8206 3134 313C 464F 75F8
```

Chad - CN: A 751 byte virus which sets the time and date fields of an infected file to 0. The tenth generation of an infected file causes a display of a person's head looking over a wall and the message: 'WOT!! No Anti-Virus Software.....' The PC then hangs.

```
CHAD            E800 0058 2D03 005F 508B F0B8 FB00 03F0 56B8 1200 03F0 8A44
```

Cod - ER: A 572 byte virus with no remarkable features.

```
Cod             FF35 FF75 021F 5F47 FF35 FF75 0283 EE04 2E8F 4402 2E8F 042E
```

Cracky - CR: The name of this virus is derived from a text string it contains. The virus makes some attempts to bypass resident monitoring programs.

```
Cracky          C3FB 80FC 4B74 2E80 FC11 7408 80FC 1274 03E9 5D01 E8E2 FF50
```

Crooked - ER: Contains one encrypted text string: ‘Only God knows!’. Awaiting analysis.

```
Crooked         32E4 B109 D3E0 91AD 3D80 FA75 08AD 3D80 7374 0DEB 153D F6C2
```

DM-400-1.04 - CR: The latest member of the DM family. A new signature string is required, as the encryption method has changed.

```
DM 400-1.04     B949 018B DE80 37?? 43E2 FAC3 BE?? ??BF 0001 57A5 A533 C08E
```

Eastern Digital - CER: 1600 byte virus which contains the text ‘MegaFuck from Eastern Digital’. The virus appears to interfere with the operation of BACKUP.COM. Awaiting analysis.

```
Eastern Digital 3D00 4B75 03EB 0F90 3D00 3D75 03EB 0790 9D2E FF2E 9B05 5550
```

Emmie - CR: A 2702 byte stealth virus. Awaiting analysis.

```
Emmie           8876 8389 4E84 33DB B8CE FACD 213D FACE 7508 83FB 0C7D 1FE8
```

End of - CR: 783 bytes. Awaiting analysis.

```
End of          F3A4 8CC2 83EA 108E DABA 0601 B021 B425 CD21 8CC8 8EC0 8ED8
```

Father - CER: This 1449 byte virus appears to be based on the Dark Avenger virus, but has been modified considerably. It contains the text ‘In memory of my father.(C)Nduk ‘91’.

```
Father          C31B D172 0429 0606 005E 561E 0E33 FF8E DFC5 069C 002E 8984
```

FCB - CEN: A 384 byte overwriting virus, probably of Russian origin. It is unusual in that it uses FCBs instead of file handles to access files.

```
FCB             BE00 02BF 8000 B980 00FC ACAA E2FC B415 BA4B 02CD 21BA 4B02
```

Funeral - CER: A Russian virus, 921 bytes long and reported to play a tune.

```
Funeral         FFE2 5EFF E650 1E06 B430 CD21 3C02 726F E8EF FF0E 1F32 C9CD
```

Globe - ER: This is a 6610 byte ‘companion’ virus which replicates in *DIET*-compressed form, similar to the ‘Even Beeper’ virus. The compression, combined with the fact that the virus itself is written in *Borland C* makes the selection of a search string problematic.

Hafenstrasse-1191 - ER: Yet another member of the Hafenstrasse family which was discovered recently ‘in the wild’ in Germany.

```
Hafen-1191      E802 0007 C31E BF00 B8B8 4000 8ED8 A049 003C 0775 03BF 00B0
```

Happy - CN: The Happy virus contains the a test string which describes its actions fairly accurately: ‘Warning !!! COM-files in current directory and C:\DOS might be infected !!!!’

```
Happy           A45E 8BEC 83EC 2CB4 1A8B D48B FA50 CD21 B44E 8D94 F800 33C9
```

Happy Monday - EN: A 7376 byte ‘companion’ virus, which claims to originate from *Lancaster Polytechnic*. Version B is very similar, but replicates in *LZEXE*-packed form and is only 5476 bytes long. The third version replicates in *PKLITE*-packed form, and is 5882 bytes long.

```
Happy Monday A  579A F909 F000 BF04 041E 579A F909 F000 89EC 5DC3 3E48 4156
Happy Monday B  E725 79AE FC77 F2F2 9FAE F814 E69A F9D9 FF6D FEF6 4BE9 C33E
Happy Monday C  5277 0EEC 9B61 0B52 1A9A F977 0793 0A01 28B5 C33E 4841 5645
```

Hellween-1182 (sic) - CER: Shorter than the first variant reported in this family, but any functional differences are not known yet.

```
Hell-1182       B440 EB03 90B4 3FE8 1600 7202 2BC1 C333 C933 D2B8 0242 EB08
```

Hi - ER: A simple, 460 byte virus. It contains the text ‘Hi’.

```
Hi              8B16 1304 4A89 1613 04B1 06D3 E2B9 4000 2629 0E02 008C C048
```

Horror-1137 - CER: The 2319 byte Horror virus reported in *VB*, June 1992 turned out to be a combination of two related viruses, 1137 and 1182 bytes long. The pattern which was given there is only valid for the 1182 byte variant, but the other one can be detected with a similar pattern:

```
Horror-1137     8BFE 83C7 0AB9 2304 2E8A 846F 042E 3005 FEC0 47E2 F8C3
```

Irus - CN: A 463 byte virus with no payload which was first reported in Estonia.

```
Irus            434B 7409 B44F CD21 72D9 4B75 F7B4 2FCD 2183 C31C 26C7 0720
```

Keypress-1232-B, Samsoft - CER: Detected with the Keypress pattern.

Kinnison - CN: This 734 byte virus is slightly polymorphic, which makes extraction of a full search string impossible. The virus contains the text ‘Dedicated to the memory of Sam Kinnison 1954-1992.’

Leprosy-Silver Dollar - CEN: A 2071 byte overwriting virus.

```
Silver Dollar   59B8 0100 EB00 5E5D C355 8BEC A157 0505 1E00 8BD0 33C9 B001
```

Lesson I - CN: A simple virus, written for ‘educational’ purposes. Does nothing but replicate.

```
Lesson I        03D6 CD21 7240 80BC C500 4D74 35B8 0242 33C9 33D2 CD21 2D04
```

Lesson II - EN: Written by the same person as the previous virus, but structurally different. The original virus was only made available in .ASM form, so the following two patterns were generated by assembling with MASM and TASM, which resulted in two different variants, 360 and 358 bytes long, but the patterns can easily be combined with the use of wildcards.

```
Lesson II-360   80BC CC00 4D75 6481 BCDE 0059 4474 5CB8 0242 33C9 33D2 CD21
Lesson II-358   80BC CA00 4D75 6381 BCDC 0059 4474 5BB8 0242 33C9 33D2 CD21
```

Little Brother-300 - ER: Very similar to the 299 byte variant, and detected with the same pattern.

Magnitogorsk-2560-C - CER: Similar to the original 2560 byte variant, but with a slightly different encryption algorithm.

```
Magnito-2560C   2E8B 851F 003D FF00 7413 BE42 0003 F7B9 BE09 2E00 042E F6AD
```

Mud - CR: This 575 byte virus is written by the authors of the ‘Swedish Boys’ viruses, and is a typical ‘Virus Exchange-BBS-only’ specimen.

```
Mud             018D 9E20 018D 96A6 013E 8A8E 0301 3BDA 7405 300F 43EB F790
```

MtE-Coffeshop - ER: This virus, which is about 3900 bytes long, uses the Mutation Engine. Anti-virus programs which detect MtE encryption should detect this virus. No search pattern is possible.

MtE-Groove - CER: A new virus which uses the Mutation Engine. It is targeted against several anti-virus products including *Norton Anti-Virus*, *Novi*, *Central Point Anti-Virus* and *Untouchable*. No search pattern is possible.

MVF - CR: This variable-length polymorphic virus contains the text ‘MAD virus Factory’, which might indicate that it has been developed with a virus construction toolkit. This virus cannot be detected with a simple search pattern.

Nov 17-768 - CER: Similar to the 855 byte variant reported earlier, and detected with the same pattern.

Old Yankee-Black Peter - CER: 1835 bytes long. Somewhat similar to the Black Wizard variant.

```
Black Peter     8CC0 8904 0E07 53B8 002F CD21 8BCB 5BBE 5B0B 81EE 0301 01DE
```

PCBB - CR: This is a group of five viruses, which seem to have somewhat variable lengths and use slightly polymorphic encryption. The variants have a base length of 1650, 1652, 1658, 1701 and 3072 bytes, and cannot be detected reliably with a search pattern. Some of the viruses crash on XT-class machines, but seem to work on ’286 and above. The viruses are later derivatives of the virus previously reported here as Plaice, but with the encryption mechanism added.

Penza - CER: This 700 byte virus uses some techniques and code fragments from the Vacsina viruses, and might be classified as a member of that family.

```
Penza           BF00 018B F281 C600 018B CB2B CEF3 A458 FA8E 57FB 8B67 F9FB
```

Pif-paf - CER: A 760 byte virus which contains the text ‘PIF-PAF B v1.0 Nincs kegyelem !’ [‘No mercy!’ Ed.]

```
Pif-paf         3DFE 4B75 04BF 0001 CF3D 004B 7403 E9E2 0156 5750 5306 521E
```

Pixel-297, **Pixel-342** - CN: Two variants that are detected with the Pixel-277 search pattern.

Plutto - CN: One of many Russian viruses reported, but not analysed this month. This one is 602 butes long and derives its name from the string ‘pLuTtoB’ which it contains. This string implies the existence of a ‘A’ variant, so far undiscovered.

```
Plutto          56BE 0000 5703 FE2E 8A05 F6D0 2E88 005F 4683 FE05 7EEE 5EB4
```

Prime - CN: A 580 byte Swedish virus, which overwrites the beginning of infected files. Awaiting analysis.

```
Prime           0130 0743 E2FB 595B 434B 740A B440 87F2 CD21 33DB EBDF C390
```

Protect - CER: Two related Russian viruses. Awaiting analysis.

```
Protect-1157    803D 4D74 1B8B F7B9 0300 BF00 01FC F3A4 5E5A 595B 5807 1F83
Protect-1355    803D 4D74 198B F7B9 0300 BF00 01FC F3A4 595B 5807 1F83 EF03
```

Quake - CEN: Related to the Ear and Suicide viruses, but only 960 bytes long and using a different encryption method. A wildcard search pattern is possible.

```
Quake           E800 00FD 5D81 ED07 018D B61E 01B9 D401 2E81 34?? 0083 C602
```

Reboot-715 - CN: A 715 byte Russian virus. Awaiting analysis.

```
Reboot-715      9006 B903 0051 31FF 8EC7 8B1C 9046 4626 8E07 90B9 FFFF B02E
```

Reboot Patcher - EN: A 5520 byte overwriting virus, written in *Pascal*. The main effect of the virus is to drop a Trojan, which is only five bytes long, but will cause the computer to reboot when the Trojanised program is executed.

```
Reboot Patcher  052A 2E65 7865 052A 2E7A 6970 052A 2E61 7263 05EA F0FF 00F0
```

Screaming Fist II-C - CER: A 692 byte variant, very closely related to the 696 byte variant reported earlier.

```
ScreamFist II-C 5D8B F556 B0?? B99F 02?? 2E30 0446 E2F9 C3
```

SHHS-B, Secret Service - CEN: A 600 byte overwriting virus, which trashes the disk and displays a message.

```
SHHS-B          01C3 BB3E 01A0 0601 0AC0 740B 3007 4302 C781 FB58 037E F5C3
```

Siskin - CER: The virus previously reported as ‘483’ or ‘Resurrect’ has now been re-classified as a member of the Siskin family. In addition, three new family members are now known, 948, 1017 and Goodbye (839 bytes long). These three viruses seem to fail miserably on some machines, and destroy all files they attempt to infect.

```
Siskin-948      48D1 E08B F88B 118A C2E6 428A C6E6 4232 E4CD 1A8B 1EA6 038A
Siskin-1017     48D1 E08B F88B 118A C2E6 428A C6E6 4232 E4CD 1A8B 1EEB 038A
Goodbye         48D1 E08B F88B 118A C2E6 428A C6E6 4232 E4CD 1ABB 8D02 8A08
```

Stahlplatte - CN: An unremarkable 750 byte virus, which does not seem to work properly on 8088-machines.

```
Stahlplatte     8EC3 BE00 00BF 0008 B900 01F3 A48E C01E E9B1 018E D8B4 47B2
```

Stanco - EN: This virus replicates in *PKLITE*-compressed form. It owerwrites the first 7529 bytes of EXE files, placing the original code at the end. Because of the high chance of false positives, no search pattern is provided for this, or any other compressed high level language virus.

Suicide - CEN: A 2048 byte virus which is closely related to the Ear virus, but uses a different encryption method.

```
Suicide         1EE8 0000 5D81 ED07 01E8 0200 EB41 B9E8 038D B634 012E 8134
```

SVC 6.0-4661 - CER: Very similar to the 4644 byte variant, and detected with the same pattern. Fully stealth.

TH-IP - CR: This 927 byte virus may be detected as a new Cascade variant by some anti-virus programs, as it uses almost the same encryption method as Cascade. Internally, the virus is quite different, however.

```
TH-IP           FAE8 0000 5B81 EB0C 018D B71F 01B9 8803 3134 310C 46E2 F9
```

Tiny Hunter - CR: A 685 byte overwriting virus, which is slightly unusual in one respect, as it does not simply overwrite the beginning of files, but places a JMP there to the actual virus code, which is located elsewhere in the file.

```
Tiny Hunter     AB8C C8AB 368E 1E2C 00BA 0800 B44B CD21 1F07 58CB CD99 CF80
```

Tired - CER: A variable-size, Russian virus. Awaiting analysis.

```
Tired           83C2 102E 0154 082E 0354 0652 2EFF 7404 1E06 5650 FCE8 0603
```

Trivial-42 - CN: Yet another attempt to write a small overwriting virus.

```
Trivial-42      B801 3DBA 9E00 CD21 93B4 40B1 2ABA 0001 CD21 B43E CD21 B44F
```

VCS-Post - CR: A minor variant of the VCS virus, with a modified encryption algorithm, but 1077 bytes long, just as the original.

```
VCS-Post        E814 008A 9C2F 058D BC20 01B9 0F04 89FE AC30 D8AA E2FA C35E
```

Vienna-415 - CN: Detected with the W13 pattern.

Vienna-744 - CN: Slightly encrypted. Detected with the GhostBalls pattern.

Vienna-Vengeance - CN: A 723 byte variant from the Phalcon/Skism virus writing group in America.

```
Vengeance       ACB9 0080 F2AE B904 00AC AE75 EEE2 FA5E 0789 7C4E 8BFE 83C7
```

Vote - CN: This 1000 byte East-European (Bulgarian?) virus does not seem to work properly. It will only append the virus code to a file, but instead of placing a JMP at the beginning, it gets written to the end!

```
Vote            AC3C 3B74 0708 C074 03AA EBF4 50B0 5CAA 1E56 0E1F E85A 005E
```

XPEH-3600, **XPEH-3608**, **XPEH-3840**, **XPEH-4048** - CER: Four new variants, similar to the 4016 byte variant reported earlier and also detected with the Yankee search pattern.

Yankee-1712 - CER: This 1716 (COM) or 1712 (EXE) byte variant of the Yankee virus appears most closely related to the 1909/1905 byte variant. Awaiting analysis.

```
Yankee-1712     7418 BE0A 0003 F3BF 0001 B920 00F3 A40E 2EFF 7746 061E 50EB
```

Yankee-2968 - CER: A 2972 (COM) or 2968 (EXE) byte variant, detected with the Yankee pattern.

DIRTY MACS

New Mac Viruses - T4-A and T4-B

A new Macintosh virus has been discovered, in two slightly different strains. The viruses were distributed in infected copies of the games program *GoMoKu* (versions 2.0 and 2.1). These infected files were posted to the Usenet *comp.binaries.mac* newsgroup, and were subsequently uploaded to a number of ftp archives, including sumex-aim.stanford.edu.

When invoked, the virus attempts to alter the System file. This alteration will be intercepted by the *SAM* anti-virus program from *Symantec* and possibly by the *Gatekeeper* public domain anti-virus software. The alert message which is displayed by the virus indicates that the *Disinfectant* anti-virus program is responsible for the alteration whether *Disinfectant* is installed on the system or not. This is an obvious deception designed to fool any user into accepting the system file modification, thus enabling the virus to continue infecting the system.

The modification of the System file results in a series of alterations to the boot code under both System 6 and System 7. The damage may render some systems unbootable but will usually result in INIT files and System extensions (respectively) not loading. The virus also attempts to modify application files on the system disk. These alterations may damage some applications by overwriting portions of the infected program with virus code. These damaged applications cannot be repaired but must be reinstalled from master software or backups.

Once installed and active, the virus does not appear to perform any obvious damage. At least one version of the virus may print a message when run after a certain number of files have become infected. The message identifies the cause of the infection as the T4 virus.

Software Updates

Authors of Macintosh anti-virus tools are planning updates to locate and/or eliminate the virus. Search data to update the principal anti-virus software is published on page 15.

Disinfectant (John Norstad), *Gatekeeper* (Chris Johnson) and *Virus Detective* (Jeff Shulman) are available from public archive sites including:

ftp.acns.nwu.edu - Northwestern University (home site of John Norstad)

microlib.cc.utexas.edu - University of Texas (home site of Chris Johnson)

sumex-aim.stanford.edu - Sumex INFO-MAC archive

rascal.ics.utexas.edu - Major Macintosh archive

This software is also available on *AppleLink*, *CompuServe*, *Genie*, *American Online*, *MacNet*, *Delphi* and via the *Usenet* news group *comp.binaries.mac*. These shareware products are regularly updated, of high quality and represent excellent value for money.

Rival, *SAM* and *Virex* are commercial products. *Rival* provides regular updates to registered users. *SAM* product updates are available from *Symantec's* BBS in the United States (408 973 9598). *Virex* product updates are available from *Microcom's* BBS in the US (919 419 1602).

ChinaTalk Trojan

A new Macintosh Trojan horse was recently discovered, called ChinaTalk which affects all Apple Macintosh computer systems. The Trojan claims to be a female sound driver which is MacInTalk compatible. The Trojan is a system extension which erases the hard disk.

Owners of *SAM* Version 3.0 can update the detection and protection capabilities of the program against this Trojan by entering the new virus definition into *SAM Virus Clinic*. In conjunction with the new *SAM User Definition* and *SAM* 3.0, the software can scan for ChinaTalk from both *Virus Clinic* and *SAM Intercept*.

Update Definition Instructions

Open *SAM Virus Clinic*. From the Options menu select 'Advanced Menus'. Select 'Add Definition (Resource)' from the Definitions menu. Enter the following information:

```
Virus Name: ChinaTalk
Resource Type: INIT
Resource ID: = 0
Resource Size: = 13392
Search String: HEX F9FA554F3F07486EFE704EBAFA12
String Offset: = 13328 FROM START
```

Search descriptions should be entered without any spaces.

As a guard against incorrect entry, *SAM* 3.0 has a 'Check' field in the definition screen. If the above information is entered correctly, the check field will equal 3453.

Once this information has been added, click 'Add' to add the definition to *SAM*. This information should be added to the *SAM* User Definitions file located in the System Folder.

(*Further update information for Macintosh anti-virus software appears on page 15.*)

KNOWN APPLE MACINTOSH VIRUSES

The following is a list of the known viruses affecting Apple Macintosh computers. Each entry includes the name (and aliases) for the virus; a short description of symptoms; together with the characteristic resources which can be used to detect the virus' presence.

Family	Name	Description
nVIR	nVIR A	When an infected application is executed nVIR A infects the system file (adding an INIT 32 resource), thereafter any reboot causes the virus to go memory-resident, after which any applications launched become infected. There is a delay before the virus announces its presence. This announcement is made once every 16 reboots or 8 infected application launches by beeping or using Macintalk to say 'Don't Panic'.
	nVIR B	Similar to nVIR A but does not utilise Macintalk. Beeps once every 8 reboot or 4 application launches.
	Hpat	All clones of nVIR B are produced by altering the resource names of the auxiliary nVIR resources created by the virus. Most anti-virus products include generic nVIR detection and can identify and disinfect such clones.
	AIDS	
	MEV#	
	nFLU	
	Jude	
	Fuck	
	nCAM	
	zero	(the resource name in this strain consists of 4 hex zero characters)
	nVIR C	Similar to nVIR B in operation. Resource patterns differ.
	nVIR ?	A forerunner to the nVIR strain. This strain is believed to delete files randomly from the system folder. nVIR A and B strains will replace this strain on infection. It is believed extinct.
Peace	DR	Also known as the Drew or MacMag virus. The virus does not infect applications but only propagates to the System file on hard or floppy disks. The virus was designed to display a message of world peace on March 2nd, 1988 and then delete itself from the System file. It is believed to be extinct.
	RR	An earlier strain with differing resource patterns.
Scores	Scores	When an infected application is run, Scores infects the system file, notepad and scrapbook files; the icons for the last two are changed to a generic document icon. Two invisible files are created, named Scores and Desktop. A reboot will cause the virus to become active in memory. Two days after infection of the system file the virus begins to infect any application run within 2 to 3 minutes of its launch. After four days any appliction with 'VULT' or 'ERIC' resources causes a system bomb (ID=12) after 25 minutes. After seven days any application with 'VULT' resources finds its disk writes returning system errors after 15 minutes of runtime.
INIT 29	INIT 29	When an infected application is run INIT 29 infects the system file and patches the open resource file trap. Any action which opens the resources file of an application or data file will cause the fork to be infected. Note that this virus does not require an application to be run for it to be infected. Only infected system files or applications will spread the virus. This virus attempts to infect any newly inserted (or mounted) disk causing the message 'This disk needs minor repairs' if it is write-protected. Sporadic printing problems may be encountered.
ANTI	ANTI A	This was the first virus on the Mac not to add new resources on infection. Instead, the virus appends its code to the CODE 1 resource of the application being infected. When an infected program is run, the virus installs itself in the system heap, and thereafter infects any application which is launched or has its resource fork opened. It does not infect the system file and only becomes active in memory when an infected application is run. ANTI does not spread under Multifinder. This virus is designed to execute automatically a code block from a disk carrying a special signature marker.
	ANTI B	A precursor strain to ANTI A. The ANTI A strain detects and modifies files infected by this strain to generate the ANTI variant (below).
	ANTI variant	A hybrid strain generated by ANTI A and ANTI B. Infected applications hang on launch.
WDEF	WDEF A	The code for this virus is stored in a WDEF (window definition code resource) in the invisible desktop file on pre-System 7 HFS volume or on MFS volumes. When a disk is inserted, all resources in the desktop resource fork are added to the search list for system resources, thus displacing the standard (innocent) WDEF in the system file. When a window is opened and the viral WDEF code is executed, 1 in 11 times the viral WDEF resource will be copied to the desktop of all mounted disks. The virus by passes anti-virus INITs by patching the trap table to call resource manipulation routines directly from ROM.
	WDEF B	This is an early debugging version of WDEF A which will beep on infection of desktop files.

Family	Name	Description
CDEF	CDEF	Using similar techniques to the WDEF virus, this simpler virus spreads by adding a viral control panel definition resource (CDEF) to the desktop file. This resource will be added to the search list for system resources in the same way as WDEF. The virus infects the desktop on all active disks. Both the CDEF and WDEF strains can be removed by rebuilding the desktop file.
MDEF	MDEF A	This virus uses a viral menu definition resource (MDEF) as the carrier. When an infected application is run, the virus changes the id of the standard system MDEF resource to 5378, adding its own MDEF 0 to the system file. Applications become infected when the menu manager executes this viral code resource. This will cause a copy of MDEF 0 to be added to the applications resource fork. The name of the added MDEF 0 provides the popular designation 'Garfield' for this virus. The virus will crash the Mac128K and 512K.
MDEF B	MDEF B	The MDEF 0 resource is named 'Top Cat' and includes code to evade detection by virus protection INITs.
	MDEF C	MDEF C contains a coding error which may cause system crashes when using the resource manager.
	MDEF D	The MDEF resource has id 8375. When the MDEF resource is executed the virus will search the last directory referenced in a file selection dialog for uninfected applications (file type = APPL).
ZUC	ZUC A	Infects applications by appending its code to the CODE 1 resource of the target file. When executed, the virus has a 1 in 4 chance of attempting to infect other applications. In most cases (15 out of 16) the application signatures in the desktop file for the volume are used to locate target applications; in 1 out of 16 cases the complete disk hierarchy will be scanned to locate target applications. The virus installs a vertical blanking interrupt task. After 90 seconds this task will cause the mouse cursor to scan diagonally across the Mac screen. The virus carries signatures of well known anti-virus products and avoids infecting such products. It also attempts to bypass protection INITs using the stored ROM addresses for key functions.
	ZUC B	This strain replaces any ZUC A strains encountered. The virus has a 1 in 2 chance of infecting an application.
	ZUC C	The ZUC C strain is capable of infecting applications which specify a CODE resource other than CODE 1 as their main code segment. ZUC C will replace any ZUC A and ZUC B strain found.
Aladin	Aladin	Reported by the *University of Hamburg* catalog project. This virus infects all Mac systems (including emulators). The virus adds a CODE resource to the infected file. After a variable delay the virus intercepts all printing operations on Mac emulators other than the *Proficomp ALADIN* emulator.
	Frankie	Variant strain of Aladin. When run on emulators other than the *Proficomp ALADIN* emulator, the strain will display a bomb and the message 'Frankie says: no more software piracy', followed by a system crash.
MBDF	MBDF A	This virus was distributed in infected versions of *Obnoxious Tetris* and the *Ten Tile Puzzle*. The virus infects applications and adds a viral MBDF 0 resource. This may cause occasional crashes, in particular when selecting items from menu bars under 7.0.1. The virus includes code to bypass early virus protection INITs.
INIT 1984	INIT 1984	The virus infects INIT startup documents at system startup. The virus will trigger if an infected system is booted on Friday 13th in 1991 or later years. Damage includes modification of file names to random strings; modification of the file creator and type to random values and deletion of 2% of files.
CODE 252	CODE 252	The strain only infects applications under the System 6 finder. Under Multifinder or under System 7 the strain infects the system file. The strain spreads between January 1st and June 5th by adding an INIT 34. After a reboot the virus becomes active and infects applications by adding a viral CODE 252 resource. If an infected program is run, or an infected system booted, between June 6 and December 31st, the following message is displayed: 'You have a virus. Ha Ha Ha Ha Ha Ha Ha. Now erasing all disks....' The virus then deletes all viral resources. The virus causes crashes under System 7 as well as on Mac 128K, 512K and XL systems.
T4	T4A	T4 was distributed in a copy of *GoMoKu* (version 2.0) The virus attempts to modify the system file INIT 31 and boot 2 resources. The altered boot code may render systems unbootable (post 7.0.1) or interfere with INIT loading. *SAM* and *Gatekeeper* erroneously indicate that *Disinfectant* is the source of the system file alteration. The virus infects applications which may be overwritten. The virus only spreads after 15th August 1992.
	T4B	A variant of T4A with a trigger date of 26 June 1992. This strain was distributed in *GoMoKu* version 2.1.
	T4 Beta	A development version of the T4 virus strains.
Hypertext	Dukakis	This virus infects hypertext stacks and includes a message urging people to vote for 'Dukakis'.
	HC	This virus is written in hypercard. When active, any uninfected home stack loaded will be infected when the virus stack is closed. The virus includes 5 audible/visual effects including: a message 'Hey what are you doing?'; after 2 minutes the German folksong *Muss I denn* is played and is repeated at 4 minute intervals; after 4 minutes the song *Behind the blue mountain* is performed; after 5 minutes two pop-up menus are displayed at minute intervals; finally 15 minutes after activation, the message 'Don't panic' is displayed.

Macintosh Viruses - Characteristic Resources

This is a table of the characteristic resources added by common Mac viruses. In the table below ‘n’ refers to the resource number of the first unused CODE resource id in the application’s resource fork. Resource name, number and length are provided. ✔ indicates that the corresponding file type is infected, and that the indicated resource will be present.

Virus	Resource	Size	System File	Application	Data file	Desktop
nVIR A	INIT 32	366	✔			
	CODE 256	372		✔		
	nVIR 0	2	✔			
	nVIR 1	378	✔	✔		
	nVIR 2	8		✔		
	nVIR 3	366		✔		
	nVIR 4	372	✔			
	nVIR 5	8	✔			
	nVIR 6	868	✔	✔		
	nVIR 7	1562	✔	✔		
nVIR B	INIT 32	416	✔			
	CODE 256	422		✔		
	nVIR 0	2	✔			
	nVIR 1	428	✔	✔		
	nVIR 2	8		✔		
	nVIR 3	416		✔		
	nVIR 4	422	✔			
	nVIR 5	8	✔			
	nVIR 6	66	✔	✔		
	nVIR 7	2106	✔	✔		
Peace RR	INIT 6	1832	✔			
Peace DR	INIT 6	1908	✔			
Scores	INIT 6	772	✔			
	INIT 10	1020	✔			
	INIT 17	480	✔			
	atpl 128	2410	✔			
	DATA 4001	7026	✔			
	CODE n+1	7026		✔		
INIT 29	INIT 29	712	✔		✔	
	CODE n	712		✔		
WDEF A	WDEF 0	1836				✔
WDEF B	WDEF 0	1842				✔
CDEF	CDEF 1	510				✔
MDEF A	MDEF 3842	314	✔	✔		
MDEF B	MDEF 8573	532	✔	✔		
MDEF C	MDEF 6982	unknown	✔	✔		
MDEF D	MDEF 8375	506	✔	✔		
CODE 252	CODE 252	1124		✔		
	INIT 34	1124	✔			
MBDF A	MBDF 0	630		✔		

Macintosh Anti-Virus Software Releases

This table provides details of recent releases of anti-virus software, together with search strings and resource information to update older releases. The author would like to acknowledge contributions from the *University of Hamburg Virus Catalog*, John Norstad and Gene Spafford.

Virus	Product	Release	User Update String
T4 strains	Disinfectant	2.9	
	Gatekeeper	1.2.6	
	Virus Detective	5.0.5	Resource CODE & Size > 3900 & Pos -1200 & WData 3F3CA9CC*31BC4E71
	Rival	1.1.9w	
	SAM		Virus Name: T4 Resource type: CODE Resource ID: Any 0 Resource size: >= 5600 Search String: Hex 2F2EFFD02F2EFFC43F3CA97B486E String offset: >= 714 from end Check value should be 'E7FA' if all search fields are entered correctly
	Virex	3.82	Guide Number = 7381312 1: 0230 FEAC 7500 00A9 / 36 2: 7B48 6EFF D62F 0E4E / BE 3: BA81 0230 FEA0 7500 / 3A 4: 00A9 7B48 6EFF D62F / 5D 5: 0E4E BA81 8280 9090 / 25
CODE 252	Disinfectant	2.8	
	Gatekeeper	1.2.6	
	Virus Detective	5.0.4	Resource Start & Size < 1200 & WData 2F2C#23F3C#2A9A0*3F3C#24878#2A9AB Filetype = ZSYS & Resource INIT & Size < 1200 & WData 2F2C#23F3C#2A9A0*3F3C#24878#2A9AB
	Rival	1.1.9v	
	SAM	3.0.8	Virus Name: C-252 Resource Type: CODE Resource ID: = 252 Resource Size: >= 1124 Search String: HEX 002248780000A9AB6100012E String Offset: = 86 FROM START Virus Name: C-252 Resource Type: INIT Resource ID: = 34 Resource Size: >= 1124 Search String: HEX 002248780000A9AB6100012E String Offset: = 86 FROM START
	Virex	3.8	Guide Number = 6324448 1: 0203 3001 7778 2A00 / 79 2: 0C50 4EFA 0003 A9AB / C4 3: 0004 A9AA 0002 A647 / B2 4: 8180 9090 9090 9090 / 1B
INIT 1984	Disinfectant	2.7	
	Gatekeeper	1.2.5	
	Virus Detective	5.0.3	Resource INIT & Size < 4500 & WData 494E#EA994*4954#8A9AB
	Rival	INIT 1984	
	SAM	3.0.7	
	Virex	3.7	Guide number: 5275840 1: 0049 4E49 5410 07C0 / 96 2: 3008 1490 7710 002F / 2C 3: 3C49 4E49 5400 0300 / 1E 4: 4AA9 AB55 4F81 8090 / 9A

© David Ferbrache, *Defence Research Agency*,
St Andrews Road, Great Malvern, UK.

TUTORIAL

Multi-Partite Viruses

Multi-partite viruses exhibit the characteristics of both boot sector and parasitic viruses. An example is Flip which infects COM and EXE files as well as the Master Boot Sector. Exploiting ‘the best of both worlds’ their chances of replication are higher than if they used only one method. It is not surprising that a few multi-partite viruses currently account for a disproportionate number of infections.

Multi-partite viruses are spread through physical exchange of any media which can be used for bootstrapping (in most cases physical exchange of floppy disks) as well as through any medium which can be used for the storage or transmission of executable code such as disks, tapes and networks. A PC is infected if bootstrapped from an infected disk or if an infected program is run.

Most multi-partite viruses such as Flip are fully multi-partite, which means that a PC infected by booting from an infected disk will infect other disks as well as programs, while a PC infected by executing an infected file will infect other programs as well as disks.

Some viruses are only partially multi-partite; for example, Spanish Telecom in a file will infect other files *as well as* boot sectors, while the same virus in a boot sector will only infect other boot sectors. The speed of propagation of multi-partite viruses is similar to parasitic viruses as they can be uploaded to bulletin boards and spread over great distances rapidly. Multi-partite viruses also spread very effectively across networks.

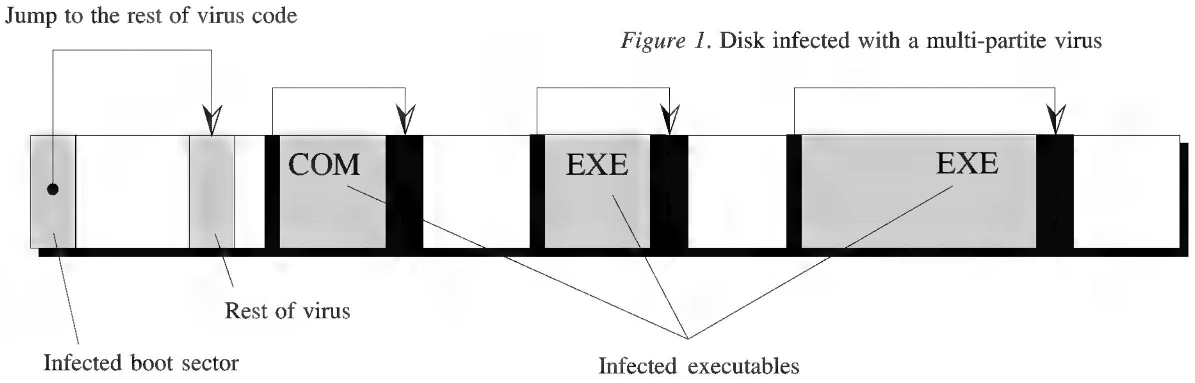


Figure 1. Disk infected with a multi-partite virus

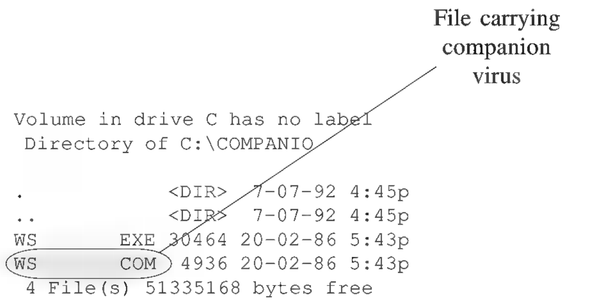
Companion Viruses

These viruses exploit the DOS property that if two programs of the same name exist in a directory, the operating system executes a COM file in preference to an EXE file.

A companion virus creates a COM file with the same name as the EXE file it ‘infects’, storing its own virus code in the COM file. When a user types in the program name, the operating system executes the COM file, which executes the virus and, in turn, loads and executes the EXE file. The directory listing in Fig. 2 shows an unsophisticated companion virus which has infected WS.EXE by creating WS.COM. More sophisticated companion viruses label the companion COM file with a DOS ‘hidden’ attribute, which means that they will not be shown in directory listings.

Note that the DOS COPY command does not copy hidden files and the virus thus denies itself the prime means of propagation: inadvertent copying of infected files by users.

Companion viruses are spread through any medium which can be used for the storage or transmission of executable code. A PC is infected if an infected program is run. It is unlikely that companion viruses will become a major threat.



```
Volume in drive C has no label
 Directory of C:\COMPANIO

.             <DIR>    7-07-92 4:45p
..            <DIR>    7-07-92 4:45p
WS        EXE 30464 20-02-86 5:43p
WS        COM  4936 20-02-86 5:43p
 4 File(s) 51335168 bytes free
```

Figure 2. A simple companion virus

PC VIRUS ANALYSIS 1

HELLOWEEN

Some ill-informed researchers and journalists seem intent upon perpetuating the myth that the former East Bloc countries are producing a race of 'super-programmers' (who thus produce 'super-viruses'). In truth, the general standard of programming is very poor and certainly does not begin to approach the excellence of professional programmers gainfully employed in the West. The fact of the matter is that the Bulgarian viruses, in particular, are simply more devious and malicious than viruses from other countries, indicating flawed characters rather than brilliant intellects.

One of the recent 'offerings' from Eastern Europe has now been reported at large in the UK. This particular virus has been called HELLOWEEN since this string is readily identified within the code. The text in Czech refers to an obscure heavy metal rock group.

Early reports suggested that the HELLOWEEN virus might have been written by the infamous Dark Avenger or that it was of Bulgarian origin. This confusion may have arisen due to the obvious 'heavy metal' allegiance of the virus writer - an enthusiasm which is shared by Dark Avenger and his cronies. However, the fact that the text message within the virus is in Czech surely points to Czechoslovakia as its country of origin, unless this is part of an elaborate deception. Dr Mike Danilak of the *Czechoslovak Institute of Anti-Virus Research* recently reported that HELLOWEEN is the second most prevalent virus in his country. Czechoslovakia recently entered the virus fray when a prolific virus writer believed to live in Bratislava released his 'Slovakia series' of polymorphic viruses.

General Description

HELLOWEEN is a resident parasitic virus which appends copies of itself to executable program files. The infection routine targets files invoked by the LOAD and EXECUTE function of DOS (4B00H) and only checks the file type internally. It is thus not possible to identify files at risk just by their extension. During testing, the virus infected EXE, COM, BIN, OVR and certain SYS files, but not COMMAND.COM for reasons which will become obvious.

Operation is quite typical and the virus has a simple trigger routine which displays a message on the screen. The virus detects its own presence using an 'RU there?' call in memory and a simple signature in files. A rudimentary attempt has been made to avoid infecting various anti-virus programs but this is clumsy and of limited effect.

Installation

The virus code is executed first when an infected program is run. After determining its own location in memory, the code checks an internal flag value which indicates the host file type. At this point, memory image files (COM type) have their initial instructions repaired to point to the correct execution point in their code.

RU There?

The virus then issues an 'RU there?' call by placing 0EC27H into AX and calling INT 21H. If the virus is resident, the call returns with 4D53H in the AX register and processing transfers to date checking routine.

It should be noted that this interrupt request is similar to one used by *Novell NetWare* and could cause unpredictable side-effects.

> *"Dr Mike Danilak of the Czechoslovak Institute of Anti-Virus Research recently reported that HELLOWEEN is the second most prevalent virus in his country."*

The installation routine continues by checking the condition of the Memory Control Block which contains the code. If the MCB is noted as the last in the chain, and if there is enough memory, the virus code is relocated to the top of memory and the MCB is modified accordingly. This has become a predictable practice with most resident viruses and makes them quite easy to detect in memory.

Once the code is installed in high memory, an INT 21H interception routine is hooked into the system and installation is complete. At this point, the date checking routine is invoked and if the date is set to November 1st (any year), processing branches to the trigger routine. At all other times, processing returns to the host program.

Trigger

If the date checking routine detects a date of 1st November, the trigger routine tests whether the current video mode is set to 80x25 text. If not, the routine aborts back to the host.

Otherwise, the screen is cleared (to red if using a colour VDU) and the following message is displayed:

```
Nesedte porad u pocitace a zkuste jednou delat neco
rozumneho!
* * * * * * * * * * * * *
!! Poslouchejte HELLOWEEN - nejlepsi metalovou
skupinou !!
```

This text translates as follows: 'Don't sit at a computer all the time. Try doing something reasonable. Listen to HELLOWEEN - the best heavy metal group!' The machine then waits until a key is pressed whereupon it will attempt a reboot which may or may not be successful. It should be noted that the message is encrypted within the virus and cannot be seen by simple inspection.

Operation

Once resident and active, the virus interception routine is fairly predictable with only two points of interest. The 0EC27H value is obviously intercepted and returned with the virus' answer. Similarly, the 4B00H (LOAD and EXECUTE) function call is intercepted in order that the target files can be infected.

This virus also intercepts calls to GET or SET the INT 21H Vector (functions 35H and 25H). Such calls have been used by some primitive anti-virus programs to detect the existence of viral activity and in this case the virus maintains a 'ghost' copy of the current INT 21H vector. Thus any attempt to recognise specific virus offsets is thwarted.

Evading Detection Software

The infection process also makes some effort to avoid alerting anti-virus software by the primitive logic of not infecting files containing certain sequences of letters in their filenames. The routine which does this is extremely inefficient and contains bugs which will affect its operation. However, in the main it does work as designed and files containing any of the following four letter groups within their name (or pathname) will not be infected:

SCAN, SHIE, TRAP, VIRU, VCOP, ASTA, ALIK, AZOR, REX., MAND, UEXE, UCOM, VIRT, CLEA, TSAF, NAV., INI., BOOT, 3P.E, LLOW

(Note that some of these groups contain a dot as one of the characters.) The last of these groups occurs as a result of one of the bugs in the virus, most of the other groups will be recognised as part of the names of various anti-virus products (*Scan*, *Shield*, *ViruTrap*, *Norton Anti-Virus* etc.). Note that the 'MAND' group prevents infection of COMMAND.COM as mentioned earlier.

During the interception, the target file is checked to see whether it is already infected; this signature is a value of 0FD71H as the last word in the file.

Conclusions

This virus is a feeble attempt at re-inventing the wheel. It is poorly designed, poorly coded and poorly executed. The usual caveats concerning so called 'benign' viruses should be observed. This virus has no deliberately damaging element built into it but it will cause system malfunction under conditions other than the trigger date. Only very primitive encryption is used to conceal the message. The code is left plain and direct recognition by a straightforward hex pattern is possible.

Acknowledgements to Dr Peter Burnett of the Bodleian Library, Oxford for his text translation.

HELLOWEEN

Aliases : None known

Type : Resident virus. Appending Parasitic on executable files (excluding COMMAND.COM)

Infection : COM type files less than 63,647 bytes, other executables of any size.

Recognition :

File If the value 0FD71H is found in the last word of a file, the virus assumes that the files is infected.

System Value of 0EC27H in AX, call INT 21H returns 4D53H in AX.

Detection : A simple hexadecimal pattern will detect this virus.

```
B440 EB02 B43F E815 0072 022B C1C3 33C9 33D2
B802 42EB 0733
```

Intercepts : INT 21H function 4B00H for infection.

INT 21H functions 2521H and 3521H to return false values.

INT 24H for internal error handling.

Trigger : Displays message in Czech to screen.

Removal : Specific and generic disinfection is possible. Replacing infected files under clean system conditions is recommended.

PC VIRUS ANALYSIS 2

Jim Bates

Palestinian (aka MSJ)

A new virus has been reported at large in the UK although it is not yet clear exactly how widespread it has become. Known as the 'Palestinian' virus, it has also been referred to as 'MSJ' and 'MS Jerusalem'. This second name will lead to confusion since the virus is absolutely no relation of the Jerusalem family of viruses and it also violates the principle of trying to avoid virus names which have been suggested by the deviants who write these things. The virus was reported on CIX by Alan Solomon in late June:

```
>>>virus/general 4701 drsolly (2736) 23 Jun92 18:01
TITLE: Virus alert
I wouldn't ordinarily issue an alert for just one
more virus. But this one is a bit different. It was
deliberately sent to a shareware vendor, and I
can't imagine why he would only send to one, so he
may have sent it to others.
```

General Description

This is a non-resident parasitic virus which prepends its code to EXE and COM files. It is non-encrypting and has an infective length of 15392 bytes. This excessive size is not an indication of complexity, but due to the fact that this program is written in a high-level language (probably Microsoft Pascal) and contains duplicate sets of internal library routines. Tests indicate that this virus infects a single file at random anywhere on drives A:, B: or C: whenever the code is executed. The virus claims to be harmless but several file types were irreparably damaged during tests.

Operation

Being prepended, the virus code loads and executes first. This completes various checks before searching drives A: B: or C: (at random) for suitable files to infect. Even though the Critical Error handling routines are intercepted during this search, the continual flashing of the drive access light on floppy drives is a sure indication that searching is under way. If there are any external device drivers attached to the floppy drives, the DOS drive prompt will appear as the drive ID is changed. During its search, the virus will select a single file (either EXE or COM, chosen by the extension only) and infect it.

After completing its operations, the virus code does not pass control back to the host program (this is too difficult for the writer to achieve in a high level language). Instead, the host code is copied to a temporary file with a unique filename and then executed as a child process of the virus code. This will mean that if an error condition arises which the virus cannot handle (of which there are plenty) the system may hang and require a cold reboot. Since the child process is therefore left incomplete, the temporary file will be left visible on the disk. The limitations of printing make it difficult to display the filename here but for reference purposes the characters are (in hexadecimal):

```
94 C7 B1 BC 90 31 A6 9B 2E 65 78 65
```

Note that the last four characters are '.EXE'.

This temporary file actually consists of the original host program code with only the date and time changed. It may thus be possible to recover valuable programs by using the virus itself to generate this 'disinfection'.

The presence of this virus will cause long delays when programs are first loaded and the disk access light will be switched on as the search routines operate.

Trigger

The trigger routine displays a series of messages (in colour on appropriate monitors) at random intervals. The frequency of the messages increases slightly if the system date is after July 1992. The spelling is preserved for posterity!

The message sequence is as follows:

First an expanding window, white on blue, opens in the middle of the screen and displays:

```
M.S Jurusalem Virus
```

Beneath this (flashing red on black) is:

```
This is a HARMLESS Virus
Do not panick this is a Harmless Virus
```

At the bottom of the screen, in white on blue appears:

```
<<<Press any key to continue >>
```

After pressing a key, the screen is cleared to white on blue and the following political message appears:

```
Do not worry this virus is designed to avoide
making any damage to your files. A free Virus
remover will be send to computer Magazines by then
30th of oct 1992 So they can supply to coustomers.
This is a demonstration of what a Palestinian Boy
can do. It is made by one of these Palestinians who
are suffering every day in their own homes because
they don't want to leave these homes. It is the
most unfair situation in the world, it is a crime
which the West has committed long time ago and
```

```
still committing it until now under the name of
PEACE. Look at the Israelis, Western and Arabic
governments. They are criminals who talks a bout
peace and freedom but they never allow them and
here are the Palestinians nation in Israel standing
in their land fighting for their own rights no
matter what happens while U.S.A., Europe and some
of the Arabic nations supporting the Israelis to
fight and finish this small nation whom Jesus was
one of them and after all this they call them selfs
Christians. It is Very easy to see this truth just
wake up and remember that one day you and your
nation are going to stand in front of the Creator
of this world to be judged on what you and your
country did to the innocent people. There is a lot
a person can do to help a nation at least by
supporting this nation. It is very easy to such a
virus to destroy your data but this is not the
manners of a good Palestinian. Our soul is light
our heart is white our mind is bright and we will
always be the same no matter what we go through.
Signature: A Palestinian teenager.

Sorry for interrupting your work
```

Political comment on such a message is out of place, but the tone and content of the message leave doubts in my mind about its authenticity. If the author is genuinely who he says he is (which is by no means certain), he damages his argument irreparably by the means he uses to broadcast it.

The implicit threat that such a virus could ‘destroy your data’ but for the kind offices of the perpetrator is particularly offensive. Contrary to the assurances of the virus writer, this virus *does* cause damage and system malfunction, particularly when using directory management software under DOS 5 and also to executable files containing appended resources.

Detection

Fortunately this virus is extremely easy to detect since every copy is identical and will be found at the beginning of infected files. The messages above may easily be seen within infected files and another indicator is the presence of a ‘signature’ number - ‘99919991999-88888888’ at the beginning of the file’s executable code in plain text, just after the ‘MZ’ header.

During infection of COM files, the prepending virus code makes such files appear to the operating system as EXE type files since they contain a valid ‘MZ’ header.

No attempt is made to hide the increase in file length and no attempt is made to retain the infected file’s original date/time stamp. Thus any competent generic anti-virus detection package should have no difficulty in detecting the virus as it attempts to spread.

Disinfection

This is also an easy virus to remove. No changes are made to the host program and in most cases it is sufficient to remove the first 15392 bytes from the file to effect a complete disinfection. However, it appears that when infecting COM files, no test is made of the COM file length prior to infection. Subsequently, although the file name remains unchanged, the ‘MZ’ header causes DOS to treat the file as if it had an EXE extension. This will result in files being damaged if the original COM file is longer than about 50 kilobytes. In such cases the file should be deleted and replaced with a clean master copy or backup.

Conclusions

Even allowing for the intricacies introduced by the high level language, the construction of this virus is still extremely messy. Along with the Italian idiot ‘Cracker Jack’, this ‘Palestinian Teenager’ ranks as one of the most inept programmers around. This virus is unlikely to cause much disruption as its operation is far too obvious.

Palestinian

Aliases : MS Jerusalem, MSJ

Type: Non-resident Parasitic file infector

Infection : Infects COM and EXE files by prepending the virus code

Infective Length : 15392 bytes

Recognition : Plaintext message may be seen in files. ASCII string ‘99919991999-88888888’ is at beginning of files.

Detection : Hex Pattern will detect this virus:

```
E872 F2E8 B7FA E8D0 F0E8 08E5 3C01 7535 BFF2
3F1E 57BF 8C1C
```

(NOTE: bytes 15 and 16 were inadvertently transposed when this pattern was published last month)

Intercepts : No intercepts except during execution.

Trigger: Displays (on a random basis) screenful of text bewailing the fate of the Palestinians. Between August and December 1992 (inclusive) the message appears more often.

Removal: The first 15392 bytes may easily be removed from an infected file. The remainder should function properly but this needs to be checked. The recommended approach is to delete infected files under clean system conditions and replace from master software or backups.

PRODUCT REVIEW 1

Mark Hamilton

Integrity Master

Integrity Master (IM) is a shareware anti-virus software package from *Stiller Research* based in Tallahassee, Florida. It is available from a number of sources in the United Kingdom - the very latest edition is always posted in *VirusForum* on *CompuServe*. It is from there that I obtained version 1.22a for review.

Self-Extracting Archive

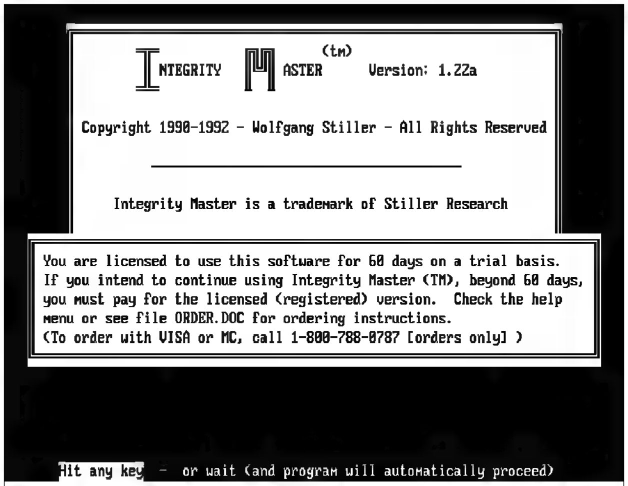
The package is distributed electronically as file IM.EXE which is a self-extracting archive. Upon entering the command 'IM', it unpacks itself into an assortment of executable and text files. I doubt the wisdom of distributing software in this way as it is vulnerable to Trojanisation and could facilitate the spread of a virus. Compressing the package into an archive is perfectly sensible, since it ensures that the user downloads the entire package and the corresponding time taken to download the file is substantially reduced. However, compressing it into a self-extracting archive (whereby untrusted code is introduced to a computer and executed) is unwise and it would be better to distribute the software in the universally accepted *PKZIP* file compression format.

Executing *IM* produces a total of 24 files, of which four are executable. There are a few miscellaneous text files which describe the product (aimed at BBS SysOps) and the activities of the *Association of Shareware Professionals*.

Installation

Installing the software is simply a case of copying the files onto a user-specified directory on the hard drive and then running a program called *SETUPIM*. This asks whether this is the first time that you are running the Setup program or whether you wish to alter an existing configuration. Answering 'Yes' displays several screens of copyright, warranty and licence information. You then have the option of running a short tutorial on the menu system used by both *IM* (the main program) and *SETUPIM*.

IM then introduces the concept of the 'needs analysis'. I've never encountered this particular expression before - basically *SETUPIM* asks a series of multiple choice questions and configures *IM* according to the user's responses (or 'needs') - just as any self-respecting configuration program should!



```
           INTEGRITY  MASTER(tm)    Version: 1.22a

 Copyright 1990-1992 - Wolfgang Stiller - All Rights Reserved

         ________________________________________

      Integrity Master is a trademark of Stiller Research

You are licensed to use this software for 60 days on a trial basis.
If you intend to continue using Integrity Master (TM), beyond 60 days,
you must pay for the licensed (registered) version.  Check the help
menu or see file ORDER.DOC for ordering instructions.
(To order with VISA or MC, call 1-800-788-0787 [orders only] )

Hit any key  -  or wait (and program will automatically proceed)
```

Shareware at its best - Wolfgang Stiller's *Integrity Master* makes many commercial products look positively amateur.

Having ascertained your level of computer literacy, security requirements and whether or not report files should be generated, *SETUPIM* looks at the disk sub-system and categorises the various drives it finds. *SETUPIM* then displays detailed instructions on completing the installation. These instructions are also written to a text file called IMPROC.TXT which can be printed out or viewed with either a text editor or the supplied document viewer, IMVIEW.COM. IMPROC.TXT details every step that should be taken right down to how to format and place the operating system onto diskettes.

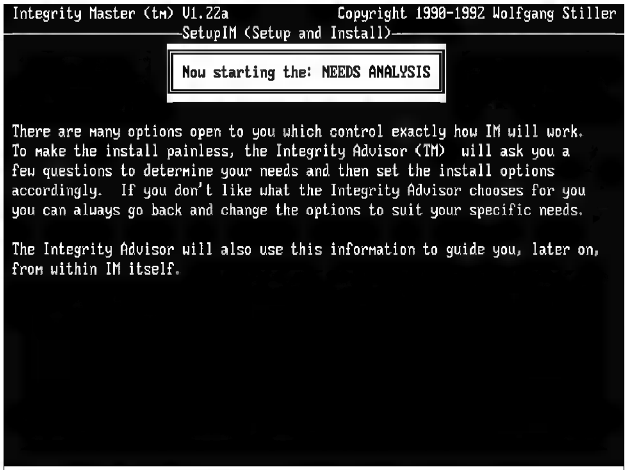
Installation complete, it's now time to fire-up the main program. *IM* first checks memory for viruses and then urges you to register the software - fair enough! Once that clears, a full-screen menu appears with an impressive array of options for virus detection and integrity checking.

Naming Confusion

A text file states that *IM* uses some 640 signatures capable of detecting a far larger number of variants. However, the names it uses do not conform to any industry standard (I would always advocate conformance with the *VB* naming convention - since every major anti-virus software developer subscribes to *VB* this is an obvious move). As an example of the confusion caused by a proprietary naming convention I cite the entry for the generic virus 1605, which *Stiller Research* calls 'Solomon' and/or 'Tel Aviv'.

Scanner Speed and Accuracy

IM's scan speeds are impressive and the program is remarkably accurate. Using the standard *VB* test set of 365 infections, it detected viruses in 346 of them - a highly



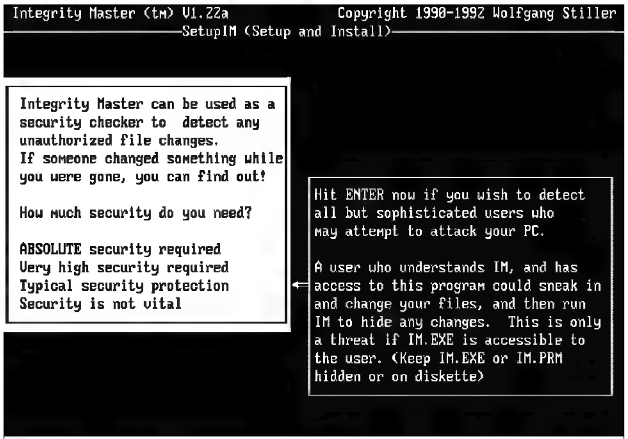
```
Integrity Master (tm) V1.22a              Copyright 1990-1992 Wolfgang Stiller
─────────────────────SetupIM (Setup and Install)──────────────────────
                    Now starting the: NEEDS ANALYSIS

There are many options open to you which control exactly how IM will work.
To make the install painless, the Integrity Advisor (TM)  will ask you a
few questions to determine your needs and then set the install options
accordingly.  If you don't like what the Integrity Advisor chooses for you
you can always go back and change the options to suit your specific needs.

The Integrity Advisor will also use this information to guide you, later on,
from within IM itself.
```

Needs testing...*SETUPIM* interrogates the user in order to configure *Integrity Master* in a manner appropriate to each individual's requirements and level of computer literacy.

creditable 95% (in a larger unofficial test set of 785 unique infections, it found 677 - an 86% detection rate). Using the 'In The Wild' test set, *IM* found all the infections barring four which contained Spanish Telecom 2. Tested againt the Polymorphic battery it found all the encrypting viruses except Mark Washburn's V2P2 and V2P6 infections. These results compare very favourably with many expensive commercial packages on offer!

Scanner Concordance

Intriguingly *IM* exposed yet another wart in *Central Point Anti-Virus* by erroneously detecting the the 'P1' virus in CPAVSCAN.DLL which forms part of that company's *Windows*-based detection software. *IM* reports 'P1' for a number of Dark Avenger's encrypting and polymorphic viruses, including Evil, Phoenix, Proud and those viruses which use Dark Avenger's Mutation Engine (MtE). However, it doesn't detect the Flip virus in *Central Point's* VSAFE or VWATCH files which other scanners have been doing ever since *CPAV* was launched. *Integrity Master* itself passes the concordance test, although several scanners (including *Total Control's VISCAN*) correctly report that *IM's* executables are compressed with *LZEXE*.



```
Integrity Master (tm) V1.22a              Copyright 1990-1992 Wolfgang Stiller
─────────────────────SetupIM (Setup and Install)──────────────────────

Integrity Master can be used as a
security checker to  detect any
unauthorized file changes.
If someone changed something while
you were gone, you can find out!        Hit ENTER now if you wish to detect
                                        all but sophisticated users who
How much security do you need?          may attempt to attack your PC.

ABSOLUTE security required              A user who understands IM, and has
Very high security required             access to this program could sneak in
Typical security protection          ⇐ and change your files, and then run
Security is not vital                   IM to hide any changes.  This is only
                                        a threat if IM.EXE is accessible to
                                        the user. (Keep IM.EXE or IM.PRM
                                        hidden or on diskette)
```

Advice and detailed information helps the user select the level of security he requires. *Integrity Master* is highly configurable and user-friendly.

Unusually, for an American anti-virus package, there is no disinfection capability - indeed, the author goes to great pains to explain just how potentially misleading and dangerous such a facility can be. How refreshing! I wish more companies were as open and honest with their users as *Stiller Research*. *IM* will repair a damaged boot sector but it does this by simply rewriting it from an encrypted copy which it takes, and saves, each time you invoke an 'initialisation' pass on a particular drive for the first time.

Generic Detection

You can check generically three classes of files in addition to the Master Boot Sector and DOS Boot Sector. There is an 'all files' option, an 'executables only' option and, somewhat confusingly, an 'all programs' option. This latter category need a little explanation. In addition to checking all your program files, it will include all program source files - such as those valuable C, Assembler and Cobol sources. On a development machine, where these source files could change from hour to hour, there'd be little point in checking these files. However, you can tailor the file extensions that are to be included so you can nominate any extensions you wish - for example spreadsheet or word processing macro files.

Stiller says that for each separate installation of *IM* a different algorithm is used to calculate the checksum values. I checked this out and the claim appears to be valid. *IM* creates a check file - named ')(.ID' - for each directory and you have the choice of storing these either in that directory or, alternatively, on a diskette. In this case, *IM* mirrors the directory structure of the drive it is checking on the diskette and places the ')(.ID' files within the appropriate sub-directories. This makes for good security and is a nice touch.

There was very little time difference between creating the initial check file i.e. the 'initialisation pass' and subsequent file checking. As the initial check takes place the subject files are scanned for known viruses.

To create the checksum values for 1,704 files occupying 64,429,215 bytes took 1 minute 52 seconds (or around half a megabyte per second).

on a 50 MHz '486. The subsequent checking pass took 1 minute 34 seconds (or nearly 600 K per second). Incidentally, included in those files was a 12 Mbyte *Windows* 3.1 enhanced mode swap file (386SPART.SWP). The time taken to create and write the check files as against reading them would account for the 18 second difference between these two timings.

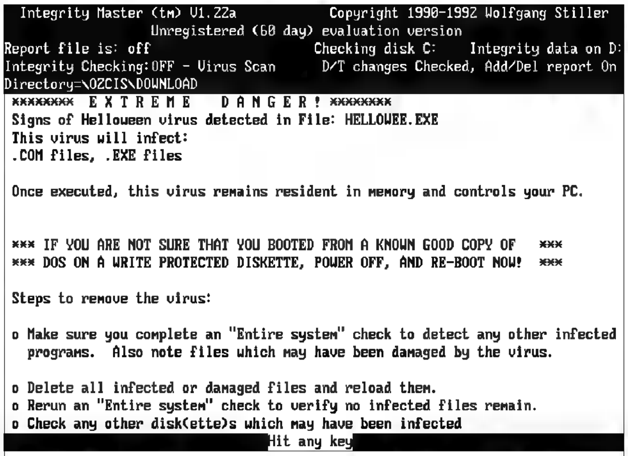
IM noticed any changes I made - including those in the middle of files, even when the file's date, time and length remained unchanged. Full marks.

Stiller has made an excellent job of documenting this product even though I would have preferred all the documentation to have been in one file. Nevertheless, it is complete with indices and a table of contents. He devotes chapters to discussing viruses and the threats that they pose as well as other ways that files can be compromised.

Conclusions

I find it difficult to fault *Integrity Master*: *Stiller Research* has done an excellent job and I am frankly amazed that the company hasn't made it a commercial package because it is vastly superior to several of the mainstream commercial anti-virus products.

Suggested improvements? First, the software should not be distributed as a self-extracting archive. Secondly, I would like to see a mechanism whereby users can add their own search patterns, rather than have to wait for the company to make an update available. I am not sure when updates are released, but judging from the dates of the various versions I found while trawling the bulletin boards, it appears that a new version is released every 6-8 weeks.



```
 Integrity Master (tm) V1.22a           Copyright 1990-1992 Wolfgang Stiller
                 Unregistered (60 day) evaluation version
Report file is: off                    Checking disk C:    Integrity data on D:
Integrity Checking:OFF - Virus Scan     D/T changes Checked, Add/Del report On
Directory=\OZCIS\DOWNLOAD
 ******** E X T R E M E   D A N G E R ! ********
 Signs of Helloween virus detected in File: HELLOWEE.EXE
 This virus will infect:
 .COM files, .EXE files

 Once executed, this virus remains resident in memory and controls your PC.


 *** IF YOU ARE NOT SURE THAT YOU BOOTED FROM A KNOWN GOOD COPY OF   ***
 *** DOS ON A WRITE PROTECTED DISKETTE, POWER OFF, AND RE-BOOT NOW!  ***

 Steps to remove the virus:

 o Make sure you complete an "Entire system" check to detect any other infected
   programs.  Also note files which may have been damaged by the virus.

 o Delete all infected or damaged files and reload them.
 o Rerun an "Entire system" check to verify no infected files remain.
 o Check any other disk(ette)s which may have been infected
                                  Hit any key
```

Integrity Master provides unimpeachable advice on disinfection - the documentation actually warns of the dangers of automatic disinfection software.

Integrity Master

Scanning Speeds

Secure Mode (All Files - 1,704)	1m 50s
Turbo Mode (Executables only - 469)	44s

Scanner Accuracy

VB Standard Test Set [1]	346/365	94.79%
Enlarged Test Set [2]	677/785	86.24%
In The Wild Test Set [3]	112/116	96.55%
Polymorphic Test Set [4]	80/150	53.33%

(Note: There was no difference in scanner accuracy between the Turbo and Secure operating modes.)

Checksumming Speeds

Secure (All Files, calculate checksums)	1m 52s
2nd pass (All Files, validate checksums)	1m 34s
Turbo (Executables, calculate checksums)	48s
2nd pass (All Files, validate checksums)	39s

Concordance Test

Passed

Technical Details

Product: *Integrity Master*

Version: 1.22a

Developer: *Stiller Research*, 2625 Ridgeway Street, Tallahassee, Florida 32310, USA. Telephone/fax numbers not supplied.

CompuServe: 72571,3352

Internet: 72571.3352@compuserve.com

UK Agents: *Nildram Software*. Tel 0494 729236
PC Independent User Group. Tel 0732 771512

Price: US$30.00. Quantity discounts and site licences available.

Requirements: IBM PC/XT/AT/PS2 or compatible with 512 k and DOS 2 or later.

Test Hardware: Scanner accuracy tests were conducted on an Apricot Qi486 running at 25 MHz and equipped with 16 Mb memory and 330 Mb hard drive. All speed tests were conducted on a Kamco '486 workstation running at 50 MHz and equipped with 4 Mb memory and 120 Mb hard drive; this machine had a total of 1,704 files occupying 64,429,215 bytes of which 469 were executable and occupied 20,790,824 bytes.

For details of the various test sets, please refer to:

[1] Standard Test Set: *Virus Bulletin* - May 1992 (p.23)

[2] This unofficial test set comprises 785 unique infections.

[3] In The Wild Test Set: *Virus Bulletin* - June 1992 (p.16)

[4] Polymorphic Test Set: *Virus Bulletin* - June 1992 (p.16)

PRODUCT REVIEW 2

Dr Keith Jackson

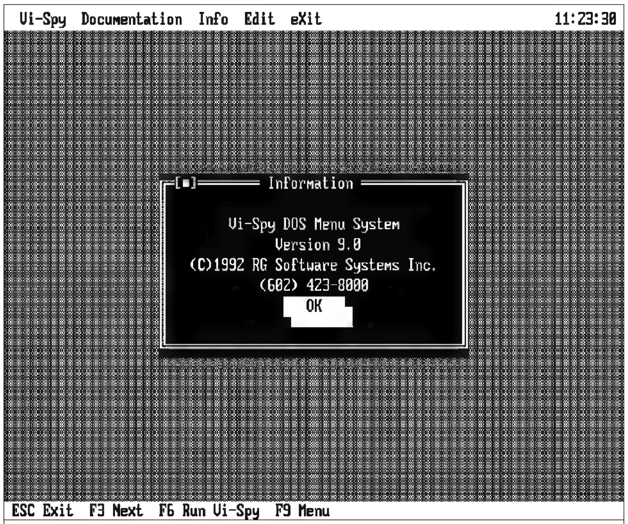
Vi-Spy - Professional Edition

It is now over two years since *VB* reviewed *Vi-Spy* (May 1990 to be precise). If a week is a long time in politics, then two years is an eternity in the development of anti-virus software, so another look at *Vi-Spy* is now long overdue.

Vi-Spy version 9 includes a host of features including an automatic scheduler program (*AUTOVS*) which conducts a scan of the system at pre-determined intervals, memory map comparison, hidden file count and list, integrity self-checking and a facility to save boot sectors. A TSR with a range of options is also included. *RG Software* refer to the term '8-in-1': *Windows*, DOS, LAN, Standalone PCs, Detection, Removal, Protection and Scheduling. The options are numerous: this review concentrates primarily on *Vi-Spy's* virus-specific detection features.

'Virus Primer'

Vi-Spy came with two A5 booklets, one of which is the 'Guide to Operations' - a 45 page long user manual. The other booklet (67 pages) is entitled the *Computer Virus Primer and Troubleshooting Guide*, which contains an excellent description of what viruses are, how to combat them, and what to do if a virus is actually detected. It also provides a very good explanation of how a PC bootstraps, and how a virus can interact with this process. I particularly like the way in which emphasis is placed on the fact that though many software packages (*Vi-Spy* included) offer a 'cleanup' facility which removes viruses from infected files, this process can never be guaranteed to work and should be used with due caution. I even learned from the booklet that the *FDISK* supplied with version 5.0 of MS-DOS can be persuaded to repair the Master Boot Sector of a hard disk without affecting the partitioning [*using the syntax* 'FDISK /MBR'. *Ed.*].


Vi-Spy Documentation Info Edit eXit
11:23:30
[■] Information
OK
ESC Exit F3 Next F6 Run Vi-Spy F9 Menu

Vi-Spy DOS Menu System
Version 9.0
(C)1992 RG Software Systems Inc.
(602) 423-8000

A menu driven *Windows* interface - a market driven response inappropriate to anti-virus software.

I think that this *Virus Primer* has been pitched at just the right level. It is difficult to explain viruses in terms understandable by non-technical PC users. Producing a 'Kiddies' Guide to Viruses' is of no use to anyone. Conversely, there is a danger of explaining things in overly-complex terms. This booklet steers a course midway between these extremes and will prove very useful to anyone using anti-virus software for the first time.

Standard Naming Convention

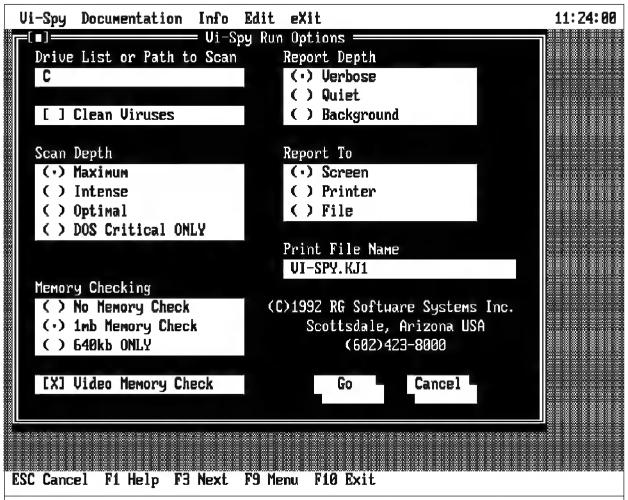
Vi-Spy is provided on both 3.5 inch (720 Kb) and 5.25 inch (1.2 Mb) floppy disks. The manual mentions that 360 Kb floppy disks (5.25 inch) are available, but only on request. Free quarterly updates are provided for one year from the date of purchase. Support is also provided via a Bulletin Board (see *Technical Details* for the phone number).

The documentation states categorically that *Vi-Spy* uses the *VB* naming convention for all viruses. There have been various attempts to standardise virus naming conventions, none of which have been successful, so it is good to see a manufacturer trying to stick to a known naming convention rather than inventing a proprietary nomenclature.

Installation

Installation to a hard disk (in any desired subdirectory) is very straightforward, with the install program simply requesting information about where the software should be installed, whether *Windows* is to be used, etc. A fast scan (memory, all boot sectors and some DOS files) is performed before installation commences. Some *Vi-Spy* files are supplied in compressed form (using *LZH* data compression), and they are automatically decompressed during installation. After installation is complete, *Vi-Spy* can either be activated as a parameter driven DOS program, or via a drop-down, mouse driven, menu interface. Either of these methods works under both DOS and *Windows*.

On-line help is provided in the form of text files which can be browsed via the drop-down menu interface. I liked the fact that all error messages are documented in a text file, thereby ensuring that they are kept up to date. This is in marked contrast to many packages where error reports are not mentioned anywhere in the documentation.



```
Vi-Spy  Documentation  Info  Edit  eXit                              11:24:00
[■]                    Vi-Spy Run Options
Drive List or Path to Scan          Report Depth
C                                   (•) Verbose
                                    ( ) Quiet
[ ] Clean Viruses                   ( ) Background

Scan Depth                          Report To
(•) Maximum                         (•) Screen
( ) Intense                         ( ) Printer
( ) Optimal                         ( ) File
( ) DOS Critical ONLY
                                    Print File Name
                                    VI-SPY.KJ1
Memory Checking
( ) No Memory Check                 (C)1992 RG Software Systems Inc.
(•) 1mb Memory Check                    Scottsdale, Arizona USA
( ) 640kb ONLY                              (602)423-8000

[X] Video Memory Check                  Go         Cancel

ESC Cancel  F1 Help  F3 Next  F9 Menu  F10 Exit
```

Running options - note the variable scan depth ranging from a scan of critical files only to a full byte-by-byte search of all files on disk. A scan report can be printed or saved to disk.

I don't think that *Vi-Spy* needs a drop-down menu interface. It's easy enough to use without such fripperies. However, the developer has deferred to the inevitable market pressure to provide this feature and its presence does no harm.

The latest version of *Vi-Spy* 'knows' about 750 unique viruses (an increase of 250 from the last major upgrade). This is in stark contrast to the version reviewed two years ago which described only 22 known viruses in the manual, and increased that number to 46 in the accompanying README file. How the world has moved on in two years! Interestingly, the manual warns 'BEWARE THE VIRUS NUMBERS GAME' - an apposite comment; in accuracy tests *Vi-Spy* has continually beaten other scanners which claim to detect many more viruses!

The original version of *Vi-Spy* requested that it should not be installed on a hard disk, but that it should always be executed directly from a write-protected floppy disk, thereby preventing the possibility of the program itself becoming infected. This is sound advice, but the addition of the menu driven front end and all the online documentation reduce the likelihood that the program will be run this way. However, the menu program does contain an option to make a 'Maintenance' disk, a diskette version of *Vi-Spy*.

Scanner Accuracy

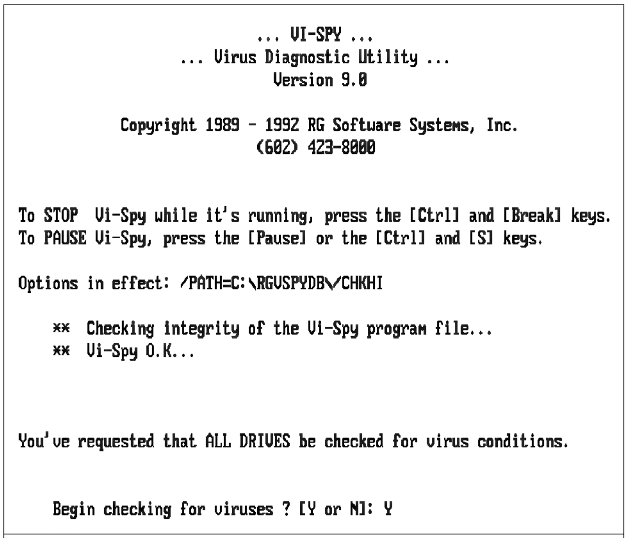
Vi-Spy was tested against the viruses listed in the *Technical Details* section. With just one exception it detected them all, no matter which scanning options were set. The exception was the Kamikaze virus, a point of academic interest only as this virus is unlikely ever to be seen in the wild. *Vi-Spy* has produced consistently good results in *VB* tests; in the most recent test (*VB*, June 92, pp. 13-16), *Vi-Spy* gained a perfect rating for its ability to detect viruses known to be in the wild and a selection of polymorphic (encrypting, self-modifying) specimens.

Scanner Speed

Vi-Spy's scanning speed was measured by searching the entire contents of a hard disk, 728 files spread across 22.7 Mbytes. The time taken by *Vi-Spy* to scan this disk took 26 seconds. For comparison purposes, *SWEEP* (v.2.39) from *Sophos*, and *Findvirus* from *Dr. Solomon's Anti-Virus Toolkit* (v.5.59) scanned this disk in 19 seconds and 15 seconds respectively. When every part of every file was scanned, *Vi-Spy's* scanning time checked in at 7 minutes 44 seconds (this is the most secure option and its use is only recommended once a virus has been detected using the scanners's 'turbo' mode). The same detection rate was measured no matter which of the scanning modes was used, so the 'turbo' mode is still efficient at detecting viruses.

Vi-Spy's test timings were *exactly* the same when the program was run under *Windows*. This is unusual since *Windows* makes programs typically run more slowly by a factor of two. I'm not sure whether this is a reflection of efficient coding in *Vi-Spy*, or the consequence of using a very fast PC for this month's testing. *Vi-Spy* was previously among the fastest scanners tested. The above figures show that it has lost some of that speed advantage. Having said this, *Vi-Spy* scan speed is perfectly acceptable.

The scanner also provides a complete screenful of information about each virus detected, with details about each



```
                        ... VI-SPY ...
              ... Virus Diagnostic Utility ...
                        Version 9.0

          Copyright 1989 - 1992 RG Software Systems, Inc.
                       (602) 423-8000


To STOP  Vi-Spy while it's running, press the [Ctrl] and [Break] keys.
To PAUSE Vi-Spy, press the [Pause] or the [Ctrl] and [S] keys.

Options in effect: /PATH=C:\RGVSPYDB\/CHKHI

   **  Checking integrity of the Vi-Spy program file...
   **  Vi-Spy O.K...



You've requested that ALL DRIVES be checked for virus conditions.


   Begin checking for viruses ? [Y or N]: Y
```

A satisfactory self-check precedes a scan of all disk drives.

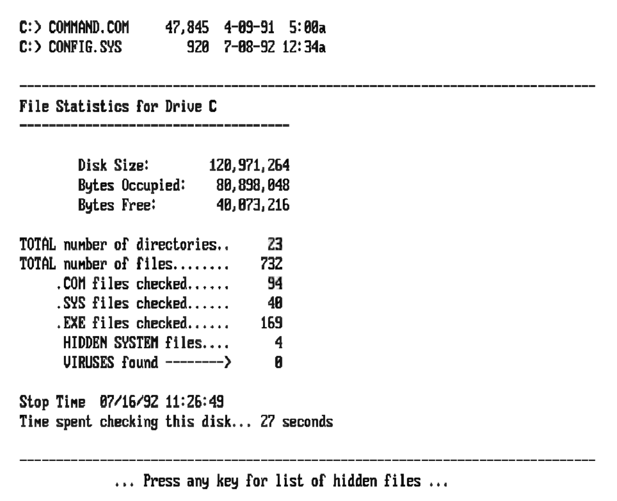
virus' infective length, the types of file or sector infected, transmission methods, associated symptoms, trigger routines and disinfection. This feature is simply excellent.

Memory-Resident Feature

A memory-resident program (RVS) is provided with *Vi-Spy*. RVS occupies 19.25 Kb of RAM and can be loaded high thus consuming no conventional memory. RVS searches files for viruses as they are accessed. Such an action imposes an inevitable overhead on system performance; in recent reviews of various anti-virus products the increase in program load/copy time has occasionally exceeded 250%!

I thus measured the overhead imposed by RVS by recording the increase in the time taken to copy 90 files (2.3 Mbytes) from one subdirectory to another, being very careful to disable any disk cache, avoid using data compressed partitions, and ensuring that the copy was made to/from exactly the same parts of the hard disk. With no memory-resident option active, this test took 23 seconds, which increased to 32 seconds when the memory-resident option was activated in its default mode. When a complete scan was used this time increased again to 36 seconds. These times represent increases of 28% and 56% respectively, a very creditable performance given the amount of checking that has gone on during the copying process.

The courteous nature of RVS revealed itself when I accidentally rebooted while it was still active, and a floppy disk had been left in drive A:. *Vi-Spy* intervened, reminded me that I was about to boot from a floppy disk and requested confirmation that this was my intention!



```
C:> COMMAND.COM     47,845  4-09-91  5:00a
C:> CONFIG.SYS         920  7-08-92 12:34a

-------------------------------------------------------------------------------
File Statistics for Drive C
-------------------------------------

        Disk Size:        120,971,264
        Bytes Occupied:    80,898,048
        Bytes Free:        40,073,216

TOTAL number of directories..     23
TOTAL number of files........    732
     .COM files checked......     94
     .SYS files checked......     40
     .EXE files checked......    169
      HIDDEN SYSTEM files....      4
      VIRUSES found -------->      0

Stop Time  07/16/92 11:26:49
Time spent checking this disk... 27 seconds

-------------------------------------------------------------------------------
             ... Press any key for list of hidden files ...
```

All clear! A comprehensive report of the directory and file structure on drive C: Note that hidden files do not escape *Vi-Spy's* attentions!.

Conclusion

Last time around, I concluded that '*Vi-Spy* is simple to understand (it detects viruses and destroys them by overwriting), easy to use, and very fleet of foot in searching for virus signatures on a disk'. Nothing has made me change that conclusion. *Vi-Spy* has kept up with the recent explosion in the total number of viruses. It now contains a *Computer Virus Primer and Troubleshooting Guide* which I can unreservedly recommend to the uninitiated user. In short, *Vi-Spy* knows exactly what it intends to do and does it extremely well.

Technical Details

Product: *Vi-Spy* (Professional Edition)

Developer and Vendor: *RG Software Systems Inc.*, 6900 E.Camelback Road, 630 Scottsdale, AZ 85251, USA, Tel. 602 423 8000, Fax: 602 423 8389, BBS: 602 970 6901.

Availability: *Vi-Spy* requires at least 150 Kb of memory. The core scanning program will operate using v.2.xx of MS-DOS, while other programs packaged with *Vi-Spy* require v.3.2 or above. *Vi-Spy* is compatible with *Windows* 3.0 and 3.1, and will operate on all major local networks.

Version Evaluated: v.9.0

Serial Number: None visible

Price: $89.95 (single copy), $149.95 (single copy with quarterly updates).

Hardware Used: A 33 MHz '486 PC, with one 3.5 inch (1.44 Mb) floppy disk drive, one 5.25 inch (1.2 Mb) floppy disk drive, and a 120 Mb hard disk, running under MS-DOS v.5.0.

Virus Test Set: 113 unique viruses spread across 182 individual virus samples comprising two boot sector viruses (Brain and Italian) and 111 parasitic viruses. Where more than one variant of a virus is included, the number of examples of each virus is shown in brackets.

1049, 1260, 1600, 2144 (2), 405, 417, 492, 4K (2), 5120, 516, 600, 696, 707, 800, 8 TUNES, 905, 948, AIDS, AIDS II, Alabama, Ambulance, Amoeba (2), Amstrad (2), Anthrax (2), Anti- Pascal (5), Armagedon, Attention, Bebe, Blood, Burger (3), Cascade (2), Casper, Dark Avenger, Datacrime, Datacrime II (2), December 24th, Destructor, Diamond (2), Dir, Diskjeb, Dot Killer, Durban, Eddie 2, Fellowship, Fish 6 (2), Flash, Flip (2), Fu Manchu (2), Hymn (2), Icelandic (3), Internal, Itavir, Jerusalem (2), Jocker, Jo-Jo, July 13th, Kamikaze, Kemerovo, Kennedy, Keypress (2), Lehigh, Liberty (2), LoveChild, Lozinsky, MIX1 (2), MLTI, Monxla, Murphy (2), Nina, Number of the Beast (5), Oropax, Parity, Perfume, Piter, Polish 217, Pretoria, Prudents, Rat, Shake, Slow, Subliminal, Sunday (2), Suomi, Suriv 1.01, Suriv 2.01, SVC (2), Sverdlov (2), Svir, Sylvia, Taiwan (2), Terror, Tiny (12), Traceback (2), TUQ, Turbo 488, Typo, Vacsina (8), Vcomm (2), VFSI, Victor, Vienna (8), Violator, Virus-101 (2), Virus-90, Voronezh (2), VP, V-1, W13 (2), Whale, Yankee (7), Zero Bug.

BOOK REVIEW

Jim Bates

Approaching Zero

A consideration for any author wishing to document the computer virus situation is what viewpoint he will adopt when creating his work. There are some excellent technical works on the subject and one or two appallingly bad examples too. *Approaching Zero*, by Bryan Clough and Paul Mungo reveals its stance in its title and lurid dust jacket picture of a balaclava-clad terrorist clutching a QWERTY keyboard. Sensationalist might be too strong a word for this book, but the final chapter certainly paints a doomsday scenario for the present computing industry.

This 242-page hardback recounts in gleeful detail some of the incidents and the personalities involved in various types of computer crime. This is an expensive book (at £14.99) for light reading and its anecdotal style somewhat trivialises a serious subject. That said, I did manage to read it at one sitting without falling asleep.

The authors are described as 'Computer Expert Bryan Clough and Journalist Paul Mungo'. The jacket note describing Clough as 'a member of the *National Computer Virus Strategy Group*' is only the beginning of the journalistic licence which this book takes. This group was not official, met only once nearly two years ago, is unlikely to meet again and was certainly not intended to be used as a recommendation on book jackets!

The authors acknowledge a variety of saints and sinners in the computer crime field. I personally found it most offensive to find my own name listed alongside the likes of Steven Gold, Robert Schifreen, Ralf Burger, Mark Washburn, Nicholas Whiteley and others who have demonstrated such breathtaking disregard for other peoples' property.

The prologue sets the scene with a highly dramatised description of a fraud being perpetrated by a thirteen year old hacker in the USA. The emotive pseudonyms that the hackers and virus writers give to themselves are used with telling effect throughout the book and only passing reference is made to the immature, deficient, schizophrenic and frequently criminal nature of their personalities.

The subsequent chapters continue the melodramatic flavour with titles such as 'Phreaking for Fun', 'Breaking and Entering' and 'Hacking for Profit'. The research has obviously been quite painstaking in most cases and the main stories recounted appear accurate. However, I did spot at least two places where the re-writing of history becomes too blatant to ignore. A section in the chapter entitled 'The Bulgarian Threat' which describes the arrival of the Nomenklatura virus in the *House of Commons* library is total fabrication, particularly where reference is made to Alan Solomon as the researcher called in to disassemble the code. (In fact the disassembly was accomplished by Joe Hirst, a former Technical Editor of the *Virus Bulletin.*) The fascinating point here is that both Clough and Mungo are fully aware of this (Mungo published an accurate account of this incident in the February 1991 edition of *GQ* magazine) and yet here they choose to tell the tale differently.

Bryan Clough's visit to Bulgaria and his meetings with some of the contributors involved certainly qualifies him to discuss the problem, but like most non-technical writers he falls into the trap of believing the propaganda that the Bulgarians are the new 'master race' of computer programmers. Sadly this distortion adds to the general air of approbation that the book gives to the criminals it describes.

The myth-making peaks with an apocalyptic epilogue where computer programs become 'uncontrollable forces' and uncounted numbers of Russian Lovechild viruses are conceived to be silently counting down to zero all over the world. This highlights the lack of the authors' technical veracity. The bibliography too provides a strange miscellany of items (limited to one per author) ranging from the odious Burger book *Computer Viruses: A High Tech Disease* to the discredited Tippett paper *The Kinetics of Computer Virus Replication*.

This is an opportunistic work, written to ride the current wave of interest in computer crime and aimed at a general interest market. While it does nothing to help the fight against such crime and it rewrites history at a whim, it *does* provide an interesting insight into some of the murkier areas of the computer underworld. However, as it can hardly be called a work of reference I would recommend waiting until it is available at a much lower price in paperback.

Incidentally - lovers of rural England should note that 'Wigston Magnum', far from being a 'misleadingly bucolic name', does not actually *exist* (unless someone is planning a new series about a trigger happy Midlands detective or they're flogging the local *Chateau Magna* in bigger bottles). At the time referred to in this book I lived in Wigston *Magna* (although I have since become genuinely bucolic at nearby Wistow Hall).

Title: Approaching Zero (242 pp.)
Authors: Bryan Clough and Paul Mungo
Publisher: *Faber & Faber*
Price: £14.99
ISBN: 0-571-16546-X

END-NOTES & NEWS

Two ***Cornell University* students have been indicted** on charges that they deliberately distributed a Macintosh Trojan horse to bulletin board systems in the United States (see *VB*, April 1992, p.28). David Blumenthal (20) and Mark Pilgrim (19) appeared before *Tompkins County Court* in June and will appear for trial later this year. If guilty, the duo faces sentences ranging from five years' probation to four years' imprisonment.

A joint **anti-virus research project** between *Edith Cowan University*, Western Australia and *Chung Ang University*, South Korea has received a grant totalling Aus$250,000 from the Australian government. Information from Professor Tony Watson, *Edith Cowan University*, Mt Lawley Campus, 2 Bradford St, Mt Lawley, 6050, W Australia. Tel (+61) 9 370 6333.

2nd International Virus Bulletin Conference, Edinburgh, 2nd-3rd September 1992. Information from Miss Petra Duffield. Tel 0235 531889, Fax 0235 559935.

The *European Institute for Computer Anti-Virus Research* has released a call for papers to be presented at its **annual conference** in Munich, December 7th-9th 1992. Abstracts should be no longer than 1500 words and the deadline for submission is September 11th. Information from *EICAR*, c/o Siemens Nixdorf AG, Dr Paul Langemeyer, Otto-Hahn-Ring-6, D-8000 München 83, Germany. Tel (+49) 89 636 82660.

Digital UK is expanding its business services response following increased demand after the Michelangelo computer virus threat and the City of London bombings. **Disaster planning and virus recovery services** are available. Tel 0734 856927.

Trend Micro Devices of Torrance, California has released version 2.0 of its ***PC RX* anti-virus software**. The software is apparently capable of detecting 1,650 viruses ('Mine's bigger than yours...' etc.). Tel 310-328-5892.

Flashback is 'a full featured **backup program**' from software house *Visionsoft* which includes such luxuries as scanning for viruses during the backup process. Sites licences cost £295 - yes, £295. Tel 0274 610503.

VyGARD is an 'indestructable **hardware device armed against virus invasion**' from US manufacturer *SYSTEM POWERHOUSE*. Stated infallibility is packed into the press release which reads like the film-script to *Terminator IV*. UK distributor is *Microlife*. Tel 0253 735979.

Sophos UK continues its series of **hands-on computer virus workshops**. Introductory and advanced sessions take place in Oxford, 8th-9th September. Tel 0235 559933.

IBM is holding a **virus management course** (FA57) and a **hands-on course** (FA58) in Warwick, 22nd-23rd September 1992. Tel 081 864 5373.

S&S International's **live virus workshop**, including hands-on experience with some real computer viruses is scheduled to take place at the *Missenden Abbey Management Centre* on 7th-8th October 1992. Tel 0442 877877.

VIRUS BULLETIN

Subscription price for 1 year (12 issues) including first-class/airmail delivery:

UK £195, Europe £225, International £245 (US$395)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Abingdon,
OX14 3YS, England

Tel (0235) 555139, International Tel (+44) 235 555139
Fax (0235) 559935, International Fax (+44) 235 559935

US subscriptions only:

June Jordan, *Virus Bulletin*, 590 Danbury Road, Ridgefield, CT 06877, USA
Tel 203 431 8720, Fax 203 431 8165

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.